

Pikonet, hoitotyön dokumentointisovellus case: Pihlajakoti

• • • • • • • • • • • • • • • • • • • •

Strömberg, Johan

Laurea-ammattikorkeakoulu
Laurea Leppävaara

Pikonet, hoitotyön dokumentointisovellus, case:
Pihlajakoti

Johan Strömberg

Johan Strömberg

Pikonet, hoitotyön dokumentointisovellus, case: Pihlajakoti

Vuosi 2010

Sivumäärä 62

Opinnäytetyöni tavoitteena oli toteuttaa Pihlajakodille, joka tarjoaa kuntoutumispalveluja aikuisille mielenterveyskuntoutujille, hoitotyön dokumentointisovellus aiemmin toteutetun määrittelydokumentin pohjalta. Sovelluksen on tarkoitus korvata Pihlajakodin nykyiset dokumentointikäytännöt ja helpottaa, nopeuttaa sekä yhtenäistää dokumentointiprosessin läpivientiä yrityksessä. Dokumentoinnin lisäksi sovelluksen tavoitteena on tarjota myös muita henkilökunnan työtä helpottavia ominaisuuksia, kuten muistutukset, osapuolten välinen tiedottaminen sekä yhteystietojen keskittyminen yhteen paikkaan.

Dokumentointisovellus on toteutettu PHP-ohjelmointikielellä sekä MySQL-tietokantaratkaisulla, ja se toimii täysin Internet-pohjaisesti käyttäjän selaimella. Sovellus ei siis vaadi toimiakseen erillisiä ohjelma-asennuksia käyttäjän puolelta ja siitä on pyritty myös tekemään mahdollisimman helppokäyttöinen, sekä toimintavarma. Toteutuksessa on huomioitu käyttäjien erilaiset tarpeet, esteettömyys, sekä tietoteknisen osaamisen väliset erot Pihlajakodin työntekijöiden keskuudessa.

Sovelluksen toiminnallisuus, sekä sen osa-alueet on toteutettu Pihlajakodin vaatimusten mukaan. Koska asiakasvaatimukset täsmentyivät sovelluksen kehittämisprosessin edetessä, opiskelijatyönä tehtyjä alkuperäisiä määrittelydokumentteja voitiin hyödyntää toteutuksessa vain soveltuvien osien. Kaikki sovellukseen toteutetut muutokset ja uudet osa-alueet on hyväksytetty toteutusvaiheessa Pihlajakodin vastuhenkilöillä.

Dokumentointisovelluksen laajuuden takia sen kehitystyö on edelleen kesken. Opinnäytetyönäni valmistunut versio sisältää Pihlajakodin kanssa erikseen sovitut avainkomponentit, jotka mahdollistavat sovelluksen täysimittaisen käytön ja testiaineiston keruun jatkokehitystä varten. Kehitystyötä tullaan jatkamaan uusina projekteinaan Laurea-ammattikorkeakoulun koordinoimana.

Avainsanat: Internet, CMS, tietojärjestelmät, ohjelmointi

Johan Strömberg

Patient document application Pikonet, case: Pihlajakoti

Year	2010	Pages	62
------	------	-------	----

The purpose of the thesis was to develop a documenting application for a company named Pihlajakoti. The Implementation is based on specification documentation, completed earlier, as its own thesis by Minna Hirvonen in 2007.

The main purpose of the application is to replace existent documenting routines in Pihlajakoti, to make patient documenting easier for the staff and to standardize entire documenting and storing processes in company. Besides documenting, the application also offers several other features for Pihlajakoti's staff such as reminders, a range of different informing tools and a centralized database for addresses within the company.

The application is implemented in the PHP language and it uses the MySQL database for information storage. The application operates in the local Intranet and can be used with any common web browser on a computer located in Pihlajakoti's office. No additional program installation is needed. The application is designed to be easy to use and reliable. The implementation also addresses those among the Pihlajakoti staff who lack computer skills.

The documenting application and its components are implemented for Pihlajakoti's needs and because of that the final version of the application has differences compared to the early specification documentation. The persons in charge of Pihlajakoti accept all differences within the documents and the final product during the implementation process.

Because of the large scale of the project, developing the application is still incomplete. The version finished for the thesis includes prearranged key components that allow full usage and testing of the application in Pihlajakoti. Implementation will continue on new projects coordinated by Laurea.

Keywords: Internet, CMS, information systems, programming

SISÄLLYS

1	Johdanto	7
1.1	Järjestelmä ja sen tarkoitus	7
1.2	Toimintaympäristö.....	8
1.3	Terminologia.....	8
2	Tavoitteet, rajaus ja menetelmät	8
2.1	Pihlajakodin vaatimuksiin perustuvat toiminnalliset tavoitteet.....	8
2.2	Omat tavoitteeni projektissa	9
2.3	Rajaukset.....	10
3	Toteutuksen prosessikaavio.....	11
4	Tietokantaratkaisu	12
5	Sovelluksen komponentit ja niiden toiminnallisuus	14
5.1	Käyttäjät	17
5.1.1	Uuden käyttäjän lisääminen järjestelmään	17
5.1.2	Käyttäjän poistaminen järjestelmästä	19
5.2	Kirjautuminen	20
5.3	Asiakkaat	21
5.3.1	Uuden asiakkaan lisääminen järjestelmään	21
5.3.2	Asiakkaan poistaminen järjestelmästä	24
5.3.3	Asiakastietojen muokkaus	25
5.4	Asiakkaan koostetiedot	25
5.4.1	Kuntoutussuunnitelma	26
5.4.2	Psyykinen vointi.....	27
5.4.3	Yksilötapaamiset	28
5.4.4	Hoitokokoukset	29
5.5	Ryhmät	30
5.5.1	Ryhmän lisääminen järjestelmään.....	30
5.5.2	Ryhmän poistaminen järjestelmästä	32
5.5.3	Asiakkaan lisääminen ryhmään	33
5.5.4	Ryhmäkohtaiset kommentit	34
5.5.5	Ryhmän aktiivisuuden määrittäminen.....	35
5.6	Raportit.....	35
5.6.1	Raportin lisääminen järjestelmään	36
5.6.2	Raportin poistaminen järjestelmästä.....	37
5.6.3	Raportin muokkaaminen	37
5.7	Muistutukset	38
5.8	Tiedotukset	39
6	Projektin yhteenveto	41
6.1	Tavoitteiden toteutuminen	42

	6.1.1	Ylläpidettävyys	42
	6.1.2	Luotettavuus	42
	6.1.3	Tarkoituksenmukaisuus	42
	6.1.4	Käytettävyys	42
	6.1.5	Toiminnallisuus	42
	6.1.6	Jatkokehitys	43
7		Sovelluksen jatkokehitys	43
8		Lähteet	44
9		Liitteet	45

1 Johdanto

Pihlajakoti, joka tarjoaa kuntoutumispalveluja aikuisille mielenterveyskuntoutujille, lähestyi Laurea-ammattikorkeakoulua projektilla, jonka lopputuotoksena syntyisi dokumentointisovellus. Sovelluksen tarkoituksena oli yhdistää Pihlajakodin dokumentointitarpeet yhden ohjelmiston alle. Pihlajakodin dokumentointi oli tähän asti toteutettu käyttäen Microsoftin tuotteita, kuten Word ja Excel. Dokumenttien tallennuspaikkana toimi paikallisen työaseman jaettu kiintolevy. Oli selvää, että projektille oli tarve ja valmis sovellus integroitaisiin osaksi Pihlajakodin työmenetelmiä. Heti alkuun myös selvisi, että projekti olisi suunnittelun ja toteutuksen kannalta haastava sekä laaja.

Sovelluksen kehittäminen aloitti vuonna 2007 Minna Hirvonen, joka toteutti opinnäytetyönään suunnitteludokumentin Pihlajakodin dokumentointisovelluksen toteutusta varten. Dokumenttia varten Hirvonen oli haastatellut Pihlajakodin henkilökuntaa, kartoittanut heidän tarpeensa sekä toiveet koskien sovellusta ja koostanut näistä tiedoista kattavan dokumentoinnin, jonka pohjalta sovelluksen toteutustyö oli määrä aloittaa.

Sovelluksen toteutus aloitettiin vuonna 2008 ja jo alkuvaiheessa selvisi, että aiemmin valmistunut toteutusdokumentti ei tulisi yksin riittämään, vaan sitä pystyi käyttämään ainoastaan apuna toteutuksessa. Aikataulua venyttivät Pihlajakodin muuttuvat tarpeet, muutokset sovelluksen toimintaympäristössä sekä aiemmin vähälle huomiolle tai jopa huomioitta jääneet asiat, kuten tietoturva, tietojen varmuuskopiointi ja sovelluksen ulkonäkö ja käyttöliittymä. Näiden lisäksi reunaehdot kavensi entisestään Pihlajakodin budjetti, joka salli sovelluksen toteuttamisen ainoastaan ilmaisilla työkaluilla ja alustoilla.

Opinnäytetyönäni syntynyt sovellus on rakennettu Pihlajakodin toiveiden mukaisesti ja eroaa siis monelta osin Hirvosen dokumentoimasta toteutussuunnitelmasta. Lopputuotos poikkeaa merkittävästi alkuperäisestä dokumentaatiosta toiminnallisuuden ja ulkonäön osalta. Kaikki sovellukseen toteutetut muutokset ja uudet osa-alueet on hyväksytetty aikanaan Pihlajakodin vastuuhenkilöillä.

1.1 Järjestelmä ja sen tarkoitus

Pihlajakodin dokumentointisovellus, joka myöhemmin sai työnimekseen Pikonet, on Pihlajakodille toteutettu selainpohjainen potilastietojen-, dokumenttien- sekä ajankäytönhallintajärjestelmä, joka toimii Pihlajakodin toimipisteissä. Sovellus vaatii toimiakseen Pihlajakodissa fyysisesti sijaitsevan työaseman.

Järjestelmän toiminnallisuus ja sen dynaamiset ominaisuudet on toteutettu kokonaisuudessaan PHP-skriptikielellä ja tietojen tallennus, sekä varastointi käyttäen ilmaista mutta parhaiten Pihlajakodin käyttötarkoitukseen soveltuvaa MySQL-tietokantaratkaisua.

Sovellus toimii palvelimella, johon on asennettu erillinen www-palvelinohjelmisto (Apache/WAMP), PHP 5- skriptien ajamiseen tarvittavat komponentit ja kirjastot sekä MySQL-tietokantasovellus. Yksi sovelluksen edellytyksistä oli myös, että se ei vaadi toimiakseen varsinaista kooditason ylläpitoa, vaan kaikki sen toiminnan mahdollistamat komponentit on automatisoitu käynnistymään palvelimen käynnistymisen yhteydessä.

Sovellus toimii henkilökohtaisilla käyttäjätunnuksilla. Käyttäjätasoa on kolme ja nämä tasot määrittävät sovelluksen käyttäjän oikeudet järjestelmän eri alueisiin. Käyttäjistä ainoastaan ylläpitäjä pystyy lisäämään, muokkaamaan sekä poistamaan käyttäjiä että heidän perustietojaan ja näkemään kaikki sovellukseen syötetyt tiedot. Sovelluksen ylläpito on Pihlajakodin pääkäyttäjälle. Ulkopuoliselle ylläpidolle ei siis ole tarvetta, pois lukien varsinaiset sovelluksen osa-alueisiin kohdistuvat päivitykset ja uudet versioinnit.

Järjestelmän tarkoituksena on helpottaa Pihlajakodin potilastietojen sekä dokumenttien hallintaa. Näiden lisäksi sovellus toimii myös resurssien varausjärjestelmänä ja mahdollistaa eri toimipisteiden ja työntekijöiden yhteystietojen varastoimisen keskitetysti yhteen paikkaan sekä antaa mahdollisuuden Pihlajakodin käytäntöjen yhtenäistämisen tietojen tallennuksessa.

Lisäksi sovelluksessa on työntekoa helpottavia ominaisuuksia, kuten henkilökohtaiset ja yleiset muistutukset, kalenteri sekä yhteystietojen hallinta. Järjestelmä on tehty alkuperäissuunnitelman mukaisesti mahdollisimman helposti lähestyttäväksi ja sen pääpaino onkin grafiikan ja uusimpien Internet-teknologioiden sijaan selkeydessä ja helppokäyttöisyydessä.

1.2 Toimintaympäristö

Sovelluksen toimintaympäristönä on Pihlajakodin Windows - palvelimella toimiva MySQL-tietokanta, Apache web-palvelin-sovellus, sekä PHP 5-palvelinlaajennus. Toisena vaihtoehtona on käyttää palvelinalustana WAMP-sovellusta, joka yhdistää kaikki edelliset komponentit yhden sovelluksen alle. Sovelluksen ohjelmakoodi, tietokannat, visuaalinen toteutus ja tietorakenne on tallennettu palvelimelle oman hakemistorakenteen alle.

1.3 Terminologia

PHP

"PHP (HyperText Preprocessor) on skriptikieli, joka on tarkoitettu palvelinpuolen ohjelmointiin. WWW-palvelimelle asennettava PHP-ohjelmisto tulkaa PHP-ohjelman. Ohjelman generoima tulos palautuu asiakkaalle (yleensä www-selaimella) esitettäväksi." (Kolehmainen 2006, 3.)

SQL

"SQL tulee sanoista Structured Query Language. Kyseessä on kieli relaatiotietokantojen käsittelyyn." (Kolehmainen 2006, 281.)

MySQL

"MySQL on hallintajärjestelmä SQL-tietokannoille ja sitä voi käyttää GNU GPL-lisenssillä. MySQL:n kehitys on jatkunut vuodesta 1995." (Kolehmainen 2006, 285.)

2 Tavoitteet, rajaus ja menetelmät

Ohjelmistoprojektin tavoitteena oli toteuttaa mielenterveyskuntoutujien hoitotyötä tekeville Pihlajakodin työntekijöille dokumentointisovellus. Sovelluksen tavoitteena on helpottaa Pihlajakodin henkilökunnan työtä ja yhtenäistää kirjaamiskäytännöt yhden sovelluksen alle sekä mahdollistaa jo syötetyn tiedon selaamisen ja haun sekä toimipisteiden välisen kommunikoinnin.

2.1 Pihlajakodin vaatimuksiin perustuvat toiminnalliset tavoitteet

Projektin lähtökohtana oli tuottaa Pihlajakodille järjestelmä, joka olisi helppokäyttöinen, mutta silti toiminnoiltaan tarpeeksi laaja kattaakseen kaikki nykyisen dokumentoinnin osat alueet. Sovelluksen tuli toimia Internet-ympäristössä siten, että kaikki Pihlajakodin toimipisteet pystyisivät käyttämään sovellusta samanaikaisesti

Pihlajakodin näkökulmasta sovelluksen toiminnalliset tavoitteet jakautuivat seuraavasti:

1. sisällön ylläpidettävyys
2. helppokäyttöisyys
3. luotettavuus
4. esteettömyys
5. laajennettavuus

Tärkeimpänä tavoitteena projektissa oli sisällön helppo ylläpidettävyys. Koska tarkoituksena oli siirtyä levyllä tallennetuista dokumenteista yhteen järjestelmään, kynnys sisällöntuottamiseen sekä sen ylläpitoon oli oltava matala. Sovelluksen käyttäjän pitää tuntea, että järjestelmä toimii paremmin kuin entiset kirjaamiskäytännöt. Ainoastaan tällä tavoin voidaan varmistua siitä, että sovellusta todella käytetään.

Edelliseen liittyen myös sovelluksen käytettävyyden ja luotettavuuden tuli olla sillä tasolla, että houkutusta vanhojen kirjaamiskäytänteiden jatkamiseen ei ole. Yksikin ylimääräinen sovelluksen kaatuminen tai hetki, jolloin käyttäjä joutuu miettimään ja kyseenalaistamaan sovelluksen toiminnallisuuden, vie pohjaa sen käytöltä.

Suunnitteluvaiheessa huomioitiin myös Pihlajakodin käyttäjien eriävät tarpeet sovelluksen visuaalisuuden ja tiedon esittämisen suhteen. Tavoitteena oli luoda sovellukseen kaksi eri visuaalista tilaa palvelemaan käyttäjiä.

Pihlajakodille oli tärkeää myös se, että sovelluksen pystyisi ottamaan käyttöön asteittain. Tämä takaa laajennettavuus oli yksi avainasioista myös Pihlajakodin näkökulmasta. Sovelluksen tuli olla sellainen, että siihen pystyy helposti toteuttamaan uusia osa-alueita ilman, että sovelluksen käyttö vaikeutuu tai estyy niiden asentamisen aikana tai sen jälkeen.

2.2 Omat tavoitteeni projektissa

Omalta kohdaltani projektin tavoitteet voidaan jakaa seuraaviin osiin:

1. ylläpidettävyys (ohjelmakoodi, sisältö)
2. luotettavuus
3. tarkoituksenmukaisuus
4. käytettävyys
5. toiminnallisuus
6. jatkokehitys

Suunnittelun lähtökohtana oli, että Pihlajakodille toteutetaan järjestelmä, jota on helppo ylläpitää niin ohjelmakoodin kuin käyttäjille näkyvän sisällön osalta. Ohjelmakoodin osalta tämä tarkoitti sitä, että pyrin toteutusvaiheessa käyttämään mahdollisimman paljon funktioita ja tällä tavoin estämään tiettyjen toiminnallisuuksien toistoa ohjelmakoodissa.

Funktiokirjaston käyttö edesauttaa sovelluksen ylläpitoa ja jatkokehitystä, mutta tuo mukanaan myös haasteita. Funktioiden käytössä korostuu hyvä suunnittelu, jonka pohjana on tulevan ohjelmarakenteen tunteminen. Funktiolähtöistä ohjelmointia haittaavat jatkuvat muutokset ja uusien ominaisuuksien tarve kesken projektin toteutusvaiheen. Tämä valitettavasti oli juuri sitä, jota tämä projekti noudatti.

Käyttäjille näkyvän sisällön osalta ylläpidettävyys tarkoittaa tavoitetasolla sitä, että Pihlajakoti pystyy ylläpitämään sovelluksessa toistuvaa tietosisältöä itse. Tällaista tietoa ovat esimerkiksi käyttäjätietojen ylläpito, toimipaikkojen ja niiden työntekijöiden yhteystiedot, resurssien nimet sekä yleinen tiedottaminen toimipaikkakohtaisesti.

Tavoitteena on, että sovelluksen valmistuttua se pyörisi itsenäisesti osana Pihlajakodin muita järjestelmiä.

Luotettavuus on ohjelmistoprojektin yksi suurimmasta haasteista, jossa korostuu huolellinen suunnittelu. Sovelluksen luotettavuus on monen osa-alueen summa, jossa tulee huomioida toimintaympäristö, käyttäjien kokemus tietojärjestelmistä, ohjelmakoodi, tietokantaratkaisu, sekä palvelinalusta. Jos yksi edellä mainituista alueista tuottaa ongelmia lopullisessa tuotantoversiossa, tavoiteltavaa luotettavuutta ei ole saavutettu.

Tarkoituksenmukaisuudella tarkoitetaan sitä, että sovelluksen suunnittelu- ja toteutusvaiheessa kiinnitetään huomiota niihin ominaisuuksiin, jotka oikeasti ovat tärkeitä ja hyödyllisiä käyttäjäryhmällä. Pihlajakodin tapauksessa näitä ominaisuuksia ovat tiedonsyöttölomakkeet ja syötetyn tiedon selaaminen järjestelmässä. Graafinen ulkoasu sekä uusimmat Internet-tekniikat ovat siis vähemmän tärkeässä osassa.

Käytettävyyden tulee olla tämän kaltaisessa sovelluksessa sillä tasolla, että se huomioi Pihlajakodin työntekijöiden erilaiset kyvyt käyttää tietojärjestelmiä. Sovelluksen hyvä käytettävyys luo pohjan sille, että sovellusta todella käytetään osana jokapäiväisiä työtehtäviä.

Toiminnallisuus ottaa kantaa siihen, mitkä osa-alueet todella ovat tärkeitä ja hyödyllisiä Pihlajakodin kokoiselle organisaatiolle. Nämä alueet toteutetaan hyvin ja muut jätetään pois tai omiksi jatkokehitysprojekteikseen.

Jatkokehitys tämän kaltaisissa, oppilastyönä tehdyissä projekteissa on haaste, joka vaatii suunnittelua sekä kattavaa ohjelmakoodin ja tietokantaratkaisun dokumentointia läpi projektin elinkaaren.

2.3 Rajaukset

Rajaan oman osuuteni toteutuksesta koskemaan sovelluksen perusrunkoa, sekä toiminnallisuutta, tietokantaratkaisun suunnittelua sekä käytännön toteutusta ennalta määritettyjen sovelluksen avain-komponenttien mukaisesti.

Käyttöliittymä

Käyttöliittymän ensimmäinen versio toteutettiin omana opinnäytetyönään Mikko Ahvenisen toimesta. Sovelluksen ulkoasu kuitenkin muuttui Pihlajakodin käyttäjien vaatimuksesta heti ensimmäisen testiversion jälkeen, joten käsittelen myös käyttöliittymää ja sen graafisia elementtejä raporttini yhteenvedossa.

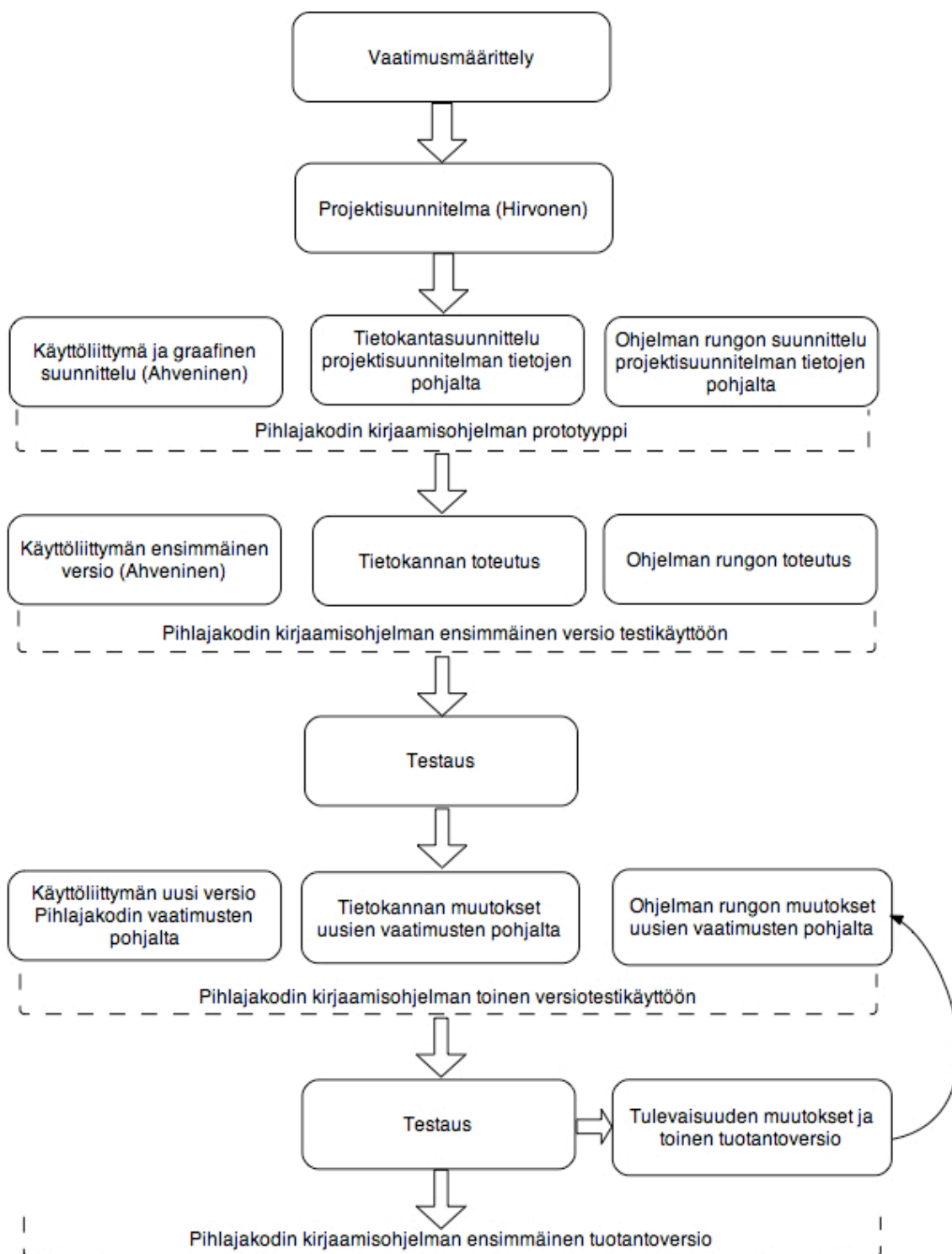
Tietoturva

Sovelluksen tietoturva, pois lukien sovelluksen käyttäjäkohtaiset kirjautumistoiminnot, tullaan toteuttamaan omina osa-kokonaisuuksinaan tulevaisuudessa. Sovelluksen toimiessa eristetyssä tietoverkossa, sitä voi käyttää ilman korkean tason salausta. Erityisen tärkeänä näen kuitenkin päivittämisen varmuuskopioinnin toteuttamisen ja syntyneen tiedon siirtämisen toiselle palvelimelle.

Tietoturvan sekä varmuuskopioiden tärkeys on tuotu esiin keskusteluissa Pihlajakodin pääkäyttäjän kanssa, joka on antanut suostumuksena nykyisen ohjelmaversion testaukseen ja käyttöön.

3 Toteutuksen prosessikaavio

Alla olevasta prosessikaaviosta selviää sovellusprojektin työvaiheet karkealla tasolla. Työvaiheet on toteutettu käyttäen perinteistä vesiputousmallia ja versiohallintaa.



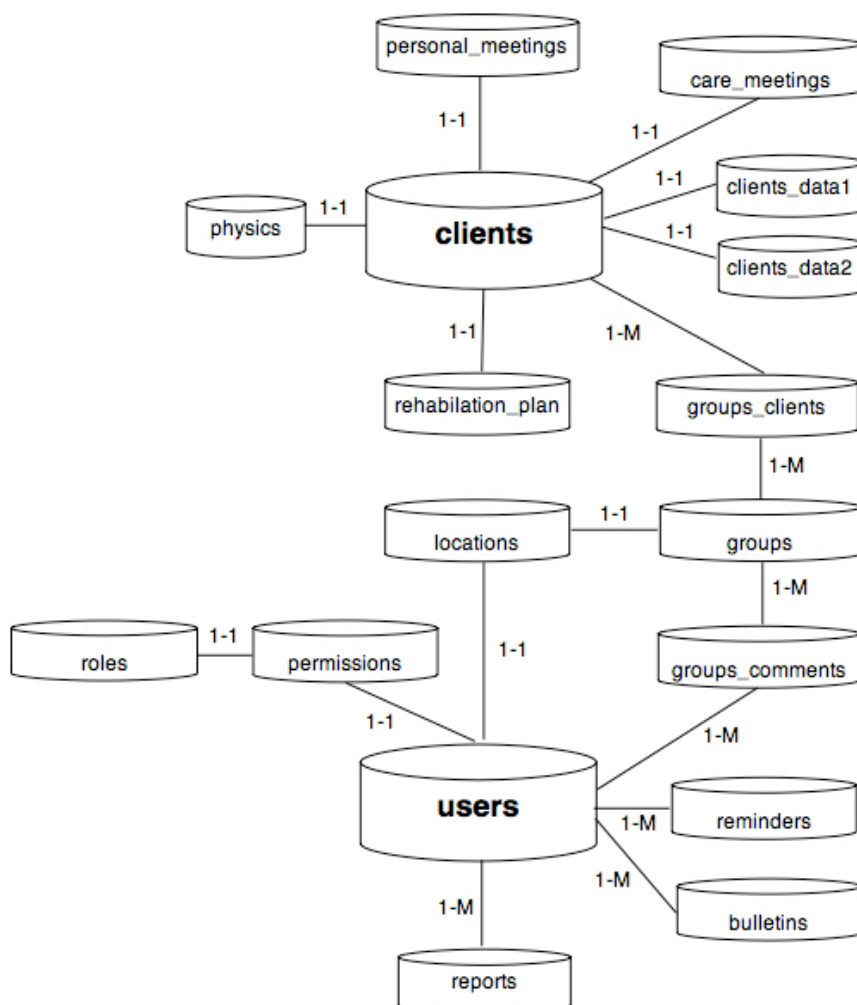
Kuvio 1. Toteutuksen prosessikaavio

4 Tietokantaratkaisu

Sovelluksen tietokantaratkaisuna toimii MySQL-tietokanta, joka soveltuu projektiin hyvin helppokäyttöisyytensä ja hyvän suorituskyvyn ja muunneltavuuden ansiosta. Dokumentointisovelluksen kautta syötetty tieto on tallennettu yhteen tietokantaan, joka jakautuu pienemmiksi kokonaisuuksiksi tauluihin.

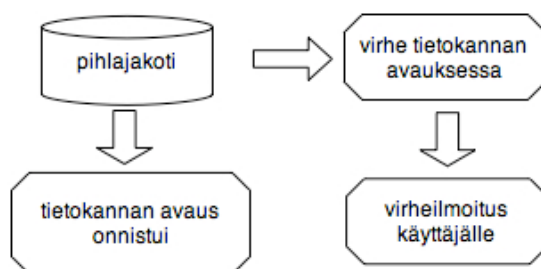
Tietokantasuunnittelussa on pyritty minimoimaan tiedon toistumista tauluissa ja näin vähentämään tietokantakyselyjen määrä minimiin. Suunnittelussa huomioitiin myös Pihlajakodin suppea käyttäjämäärä sekä verrattain pienet päiväkohtaiset määrät tietokantahauissa. Tästä syystä kantaa ei optioimu äärimmilleen, vaan suunnittelu on lähtenyt juuri Pihlajakodin tarpeista.

Tietokannan suurin yhdistävä tekijä on asiakas (clients), joka myös toimii pääavaimena sovelluksessa. Asiakkaaseen linkittyy suuri määrä tietokannan tauluja, jotka ovat riippuvaisia asiakkaan olemassa olostä tietokannassa. Toisena avaimena toimii yksittäinen käyttäjä (users) johon linkittyy asiakkaan tapaan useita tauluja. Alla olevassa kaaviossa on piirrettynä tietokannat sekä niiden suhteet.



Kuvio 2. Tietokannan taulut ja niiden suhteet

PHP:n ja MySQL:n välillä on tietokantaliittymä (database_connection). Tietokantaliittymän kutsu toistuu jokaisessa alla kuvatussa prosessissa.



Kuvio 3. Tietokantaliittymän (database_connection) prosessikaavio

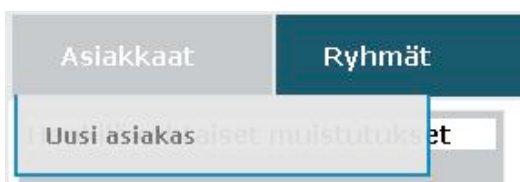
5 Sovelluksen komponentit ja niiden toiminnallisuus

Komponenttien jako on tehty karkeasti siinä järjestyksessä kuin ne jakautuvat käyttöliittymässä. Listalla olevat ”Toimintaterapia”, sekä ”henkilökunta” - komponentit on rajattu toteutukseni ulkopuolelle, enkä käsittele niiden toiminnallisuutta dokumentissani.

Käyttäjät

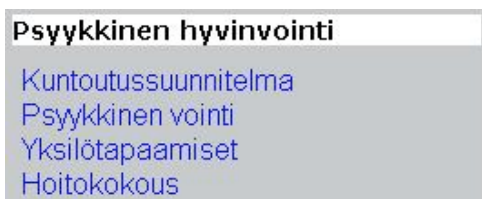
Käyttäjätietojen ylläpito

Asiakkaat



Asiakastietojen lisääminen, muokkaaminen, sekä poistaminen järjestelmästä

Asiakkaan koostetiedot



Henkilökohtainen kuntoutussuunnitelma
 Psyykkisen voinnin raportointi
 Yksilötapaamiset
 Hoitokokoukset

Ryhmät



Ryhmien lisääminen, poistaminen, sekä ylläpito
 Ryhmäkohtainen raportointi

Toimintaterapia

Toimintaterapia	Raportit
Yksilö	
OPHI	
OPHII	
Muu arvio	
Yhteenveto	

- Yksilö
- OPHI
- OPHII
- Muu arvio
- Yhteenveto

Raportit

Raportit	Toimipisteet
UUSI RAPORTTI	ARKISTOIDUT
Viikonloppuraportit	
Muut raportit	

- Viikonloppuraporttien lisääminen, muokkaaminen, sekä poistaminen
- Vapaatekstiraporttien lisääminen, muokkaaminen, sekä poistaminen

Resurssit

Resurssit	Henkilökunta
Varaa resursseja	
Resurssivaraukset	

- Resurssivaraukset, sekä niiden esikatselu ja poistaminen

Henkilökunta

Henkilökunta

- henkilökunnan yleiset asiat, kuten kehityskeskustelut

Asetukset



- Yleiset asetukset
- Muistutukset järjestelmässä
- Pääkäyttäjän hallintakonsoli

Hallintakonsoli



- Käyttäjien hallinnointi ja oikeudet järjestelmässä
- Yleinen tiedotus
- Toimipaikkojen hallinta
- Resurssien hallinta

Edellä listatut komponentit on suunnitteluvaiheessa kirjattu sovelluksen ”avainkomponenteiksi” ja ovat niitä, jotka ovat myös sovelluksen ensimmäisen tuotantoversion kannalta todettu kriittisiksi ensimmäisen vaiheen testausta ja käyttöä varten.

5.1 Käyttäjät

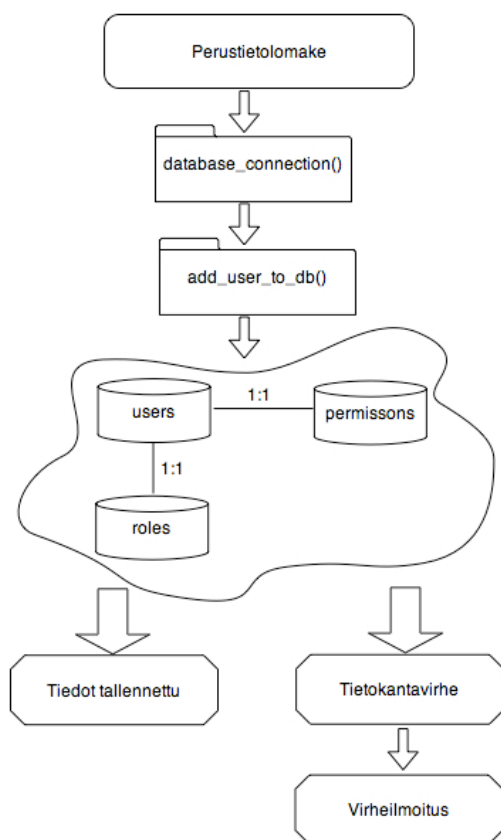
Dokumenttisovellukseen on mahdollista lisätä, poistaa käyttäjiä ja ylläpitää muuttuvia käyttäjätietoja. Käyttäjä on yksittäinen Pihlajakodin työntekijä.

5.1.1 Uuden käyttäjän lisääminen järjestelmään

Pihlajakodin pääkäyttäjällä on oikeudet lisätä järjestelmään uusia käyttäjiä. Jokaiselle käyttäjälle muodostetaan yksilöivä tunnus ja salasana, joita ilman sovellusta ei pysty käyttämään. Käyttäjätunnus muodostuu automaattisesti etu- ja sukunimen ensimmäisistä kirjaimista (yhteensä kuusi merkkiä). Salasan muodostus tapahtuu satunnaisgeneraattorilla, joka on ohjelmoitu järjestelmään omana funktionaan. Salasana oli aluksi tarkoitus lähettää uudelle käyttäjälle suoraan sähköpostiin, mutta tämä osoittautui ainakin tässä vaiheessa ongelmaksi, koska se vaatisi Pihlajakodilta erillisen sähköpostipalvelimen käyttöä.

Uusi käyttäjä

Etunimi	<input type="text"/>
Sukunimi	<input type="text"/>
Sähköposti	<input type="text"/>
Toimipaikka	Valitse toimipaikka 
Rooli	Valitse Rooli 
<input type="button" value="Tallenna tiedot ja luo käyttäjätunnus"/>	



Kuvio 4. Uuden käyttäjän lisääminen järjestelmään prosessikaaviona

Käyttäjään linkittyvien tietokantataulujen rakenne.

Kentän nimi	Tyyppi	Pituus
Id	int	11
Firstname	varchar	255
Lastname	varchar	255
Location	int	11
Phone	varchar	20
Address	varchar	100
Postno	int	6
City	varchar	20
Startpage	int	11

Taulukko 1. Users-tietokantataulun rakenne

Kentän nimi	Tyyppi	Pituus
id	int	11

user_id	int	11
username	tinytext	
password	text	
role	int	11

Taulukko 2. Permissions-tietokantataulun rakenne

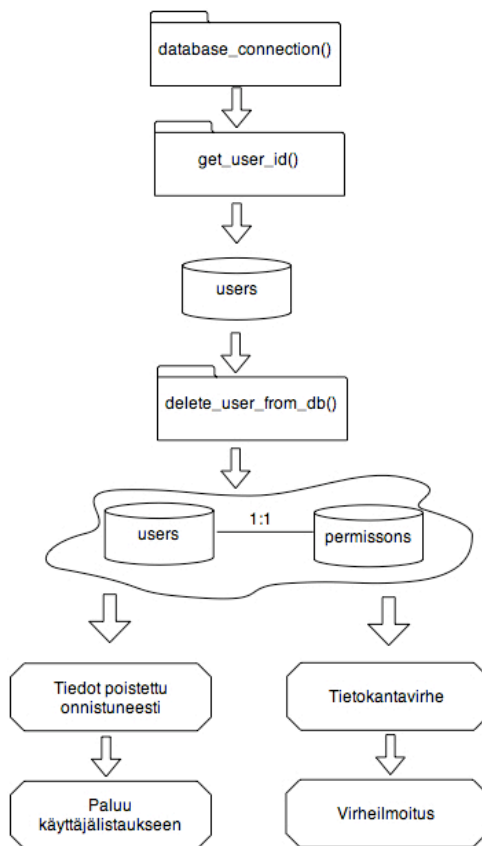
5.1.2 Käyttäjän poistaminen järjestelmästä

Käyttäjätunnusten lisäämisen tapaan myös poistamiseen tarvitaan pääkäyttäjän oikeudet. Käyttäjän voi poistaa erillisestä ylläpitäjän hallintakonsolista josta sovellukseen lisätyt käyttäjät löytyvät listattuna. Käyttäjän poistamisen myötä poistuvat myös muut häneen viittaavat tiedot tietokannasta.

Strömberg	Johan	muokkaa	poista
Testaaja	Tero	muokkaa	poista
Tavallinen	Taru	muokkaa	poista

Lisää uusi käyttäjä

Sovelluskuva 1. Käyttäjälistaus



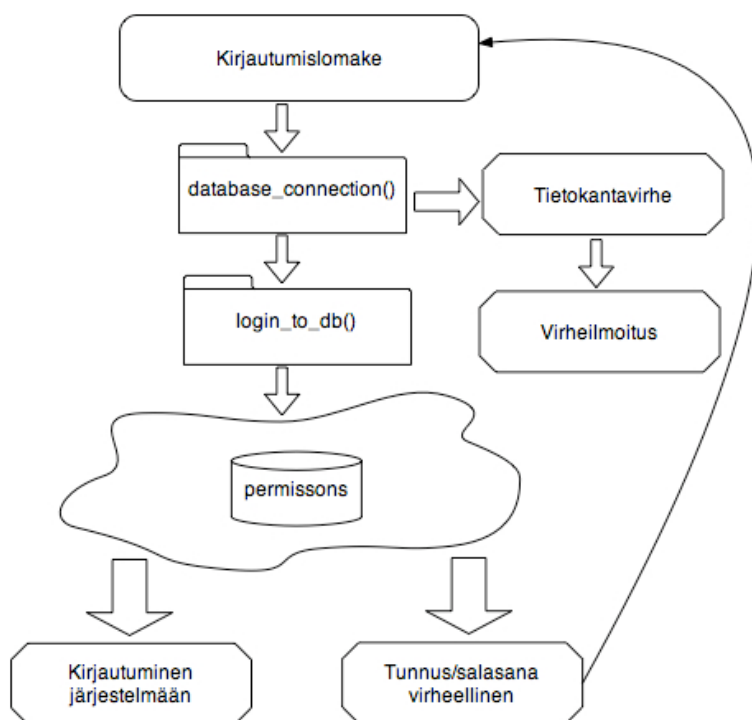
Kuvio 5. Käyttäjän poistaminen järjestelmästä prosessikaaviona

5.2 Kirjautuminen

Järjestelmään kirjautuminen vaatii käyttäjältä henkilökohtaisen käyttäjätunnuksen ja salasanan. Virheellisestä kirjautumisyrityksestä käyttäjä ohjataan takaisin kirjautumissivulle. Kirjautumisyritysten määrä järjestelmässä ei ole rajoitettu, eli tunnukset eivät lukkiudu virheellisten yritysten myötä. Tietoturva ja salaus on rajattu opinnäytetyöni ulkopuolelle. Tästä syystä kirjautumisessa käytetään heikkoa salausta.

Kirjautumisen lisäksi samalta sivulta valitaan fontin koko. Valittavana on kaksi eri fonttikokoa, normaali, sekä suuri. Valittu fontti pysyy voimassa, kunnes käyttäjä kirjautuu ulos sovelluksesta. Pääkäyttäjän toiveiden mukaisesti oletuskirjasinkoko on suuri.

Järjestelmäkuva 2. Sovelluksen sisäänkirjautumisikkuna



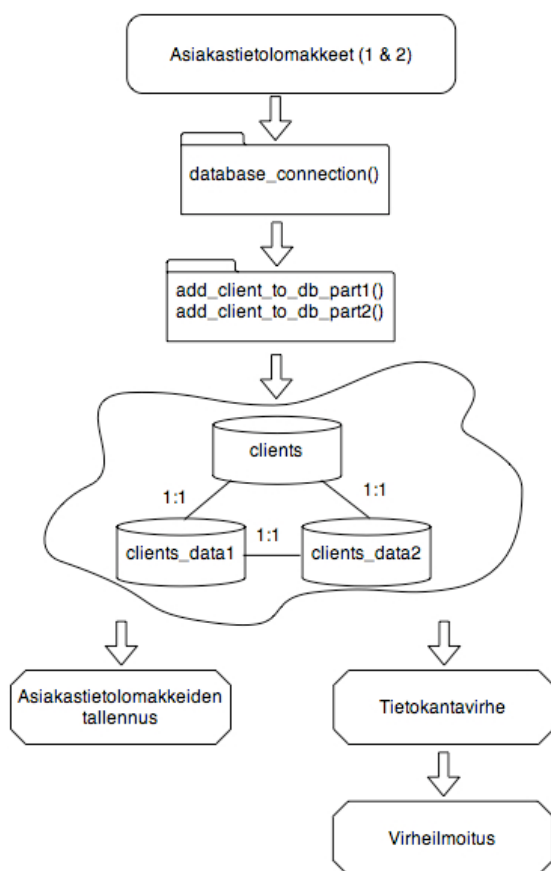
Kuvio 6. Kirjautumisen etenemisen prosessikaaviona

5.3 Asiakkaat

Asiakas toimii järjestelmän avaimena, jonka ympärille suurin osa sovelluksen komponenteista on linkitetty (pois lukien Pihlajakodin henkilökunnan käytössä olevat alueet). Tietokanta on toteutettu siten, että asiakkaaseen linkittyvät tiedot poistetaan asiakastietojen poistamisen yhteydessä automaattisesti.

5.3.1 Uuden asiakkaan lisääminen järjestelmään

Pikonetin peruskäyttäjä voi lisätä uuden asiakkaan järjestelmään. Asiakkaan lisäämiseksi käyttäjän tulee täyttää kaksiosainen asiakastietolomake. Asiakas saa lomakkeen tallennuksen yhteydessä yksilöivän, juoksevan numerotunnisteen (id), jota käytetään muissa tietokannan tauluissa linkittämään syötetty tieto oikeaan asiakkaaseen.



Kuvio 7. Asiakkaan lisääminen tietokantaan prosessikaaviona

Asiakastietotaulut Pikonet-järjestelmässä.

Kentän nimi	Tyyppi	Pituus
Id	int	11
Firstname	varchar	255
Lastname	varchar	255
social_securityno	varchar	15
Phoneno	varchar	20
Checkindate	varchar	16
Comingfrom	varchar	100
Sender	text	
Language	varchar	20
Flat	varchar	50
Address	varchar	255
Occupation	varchar	10
Employee	varchar	255
Student	varchar	11
College	varchar	255
Interests	varchar	1000
Finances	varchar	11
financecontroller	varchar	255
Benefit	varchar	11
benefitcontrollerphone	varchar	255
Relatives	varchar	11
Relativephone	varchar	25
circumstances	varchar	25
Kids	varchar	11

Taulukko 3. Clients_data1-tietokantataulun rakenne

Kentän nimi	Tyyppi	Pituus
client_id	int	11
siccnesses	text	
sleepinghabits	text	
eatinghabits	text	
medication	text	
cabable	text	
support	text	
physics	text	
goals	text	
substances	text	
substancedesc	text	
substanceuse	text	
futureplans	text	
otherinfo	text	

Taulukko 4. Clients_data2-tietokantataulun rakenne

5.3.2 Asiakkaan poistaminen järjestelmästä

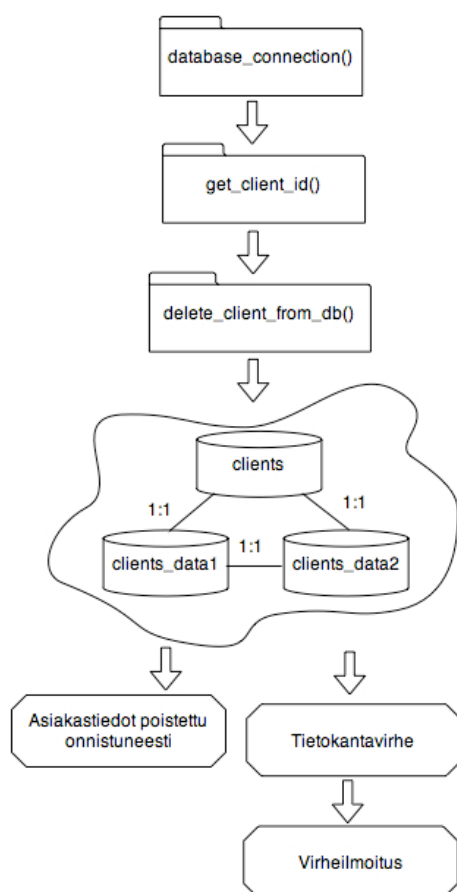
Lisätyn asiakkaan voi poistaa järjestelmästä. Asiakastietojen poistamista ei ole rajoitettu käyttöoikeuksien mukaan, vaan jokaisella käyttäjällä on mahdollisuus poistaa asiakas tietokannasta huolimatta siitä, onko käyttäjä lisännyt asiakkaan itse.

Asiakastietojen poistamisen yhteydessä sovellus varmentaa toiminnon käyttäjältä, jonka jälkeen tiedot poistetaan tietokannasta pysyvästi. Tietokanta on suunniteltu siten, että myös kaikki muu asiakkaaseen linkittyvä tieto poistetaan tietokannasta poiston yhteydessä. Tuhotuja tietoja ei ole mahdollista palauttaa nykyisessä versiossa, jossa tietokannan tietoja ei kopioida toiseen kantaan.

ASIAKKAAT		
Strömberg Johan	Asiakkaan tiedot	poista

kuva 5.4 asiakastietojen listaus

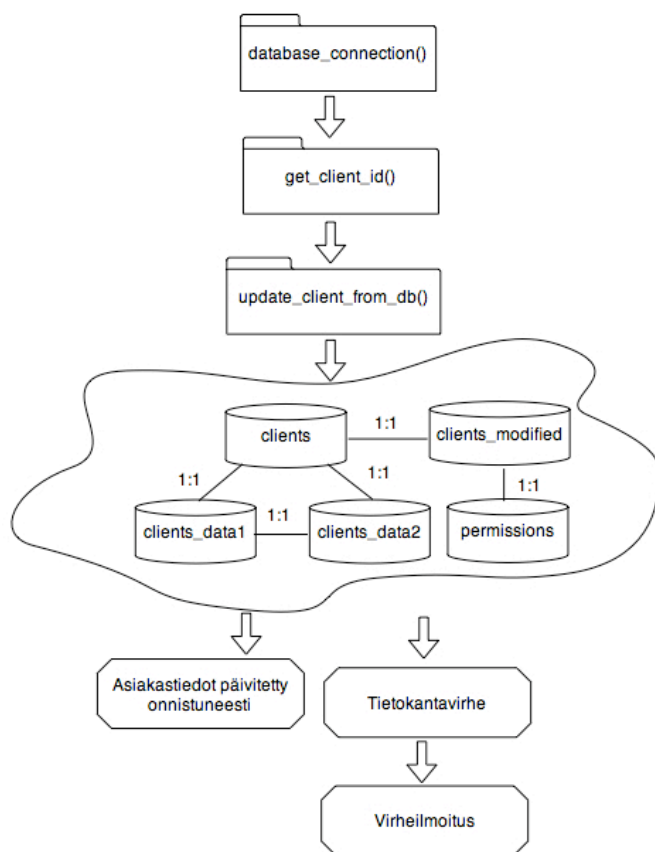
Järjestelmäkuva 3. Asiakastietojen listaus



Kuvio 7. Asiakkaan poistaminen tietokannasta prosessikaaviona

5.3.3 Asiakastietojen muokkaus

Käyttäjällä on oikeudet muokata jokaista järjestelmään lisättyä asiakasta.



Kuvio 8. Asiakastietojen päivittäminen prosessikaaviona

5.4 Asiakkaan koostetiedot

Asiakkaan perustietojen lisäksi sovelluksessa on mahdollista dokumentoida asiakkaan alle tietoja eri osa-alueilta. Näitä ovat mm. Psykkinen, sekä fyysinen hyvinvointi, henkilökohtaiset tapaamiset, sekä asiakasta ja hänen hoitoaan koskevat suunnitelmat, sekä lääkitys.

Linkit asiakkaan koostetietoihin listataan asiakastieto-sivun vasemmassa reunassa, josta tietoja on helppo seurata ja päivittää. Mikäli tietoja ei löydy järjestelmästä asiakkaan alta, avautuu käyttäjälle tyhjä sivun tiedon luomista varten.

Psyykkinen hyvinvointi	Etunimi	Johan
Kuntoutussuunnitelma	Sukunimi	Strömberg
Psyykkinen vointi	Henkilötunnus	100982-112E
Yksilötapaamiset	Matkapuhelin	0503522823
Hoitokokous	Tulopäivä	17.08.2009
	Mistä tullut?	Helsinkis
	Lähettävä taho	Helsingin kaupunki
	Äidinkieli	suomi
	Oma asunto	on
	Osoite	Kylätie 4 a 12 00320 Helsinki
	Asema	Työssä

Järjestelmäkuva 4. Asiakkaan koostetiedot

Koska ensimmäisessä tuotantoversiossa on ainoastaan osa sovelluksen määrittelyssä olleista koostetiedoista, on kaikki tiedot pyritty toteuttamaan samalla kaavalla niiden helpon ylläpidettävyyden takia. Tapaamiset ja kokoukset eroavat muista asiakkaan koostetiedoista siten, että niihin on mahdollista kerryttää historiatietoa edellisistä kerroista, jolloin asiakkaan alle syntyy helposti seurattavaa historiatietoa.

Yksilötapaamiset- Johan Strömberg

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

Tulosta dokumentti | Muokkaa dokumenttia

Yksilötapaamiset päivämäärän mukaan ▾

Yksilötapaamiset päivämäärän mukaan

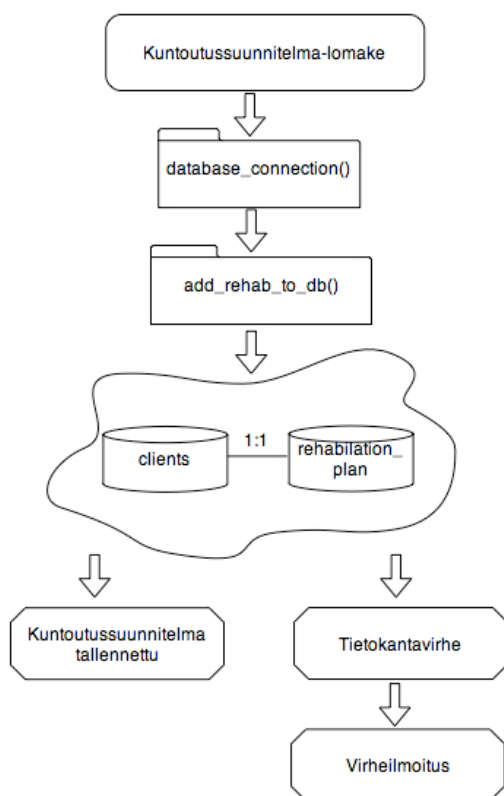
2009-12-06

2009-12-06

Järjestelmäkuva 5. Asiakkaan yksilötapaamiset

5.4.1 Kuntoutussuunnitelma

Asiakkaalle luodaan yksilöllinen kuntoutussuunnitelma. Kuntoutussuunnitelma pitää toteutuksen näkökulmasta sisällään vapaatekstikenttiä henkilön nykyisistä voimavaroista, sekä eri aikavälin tavoitteista. Kuntoutussuunnitelma linkittyy automaattisesti lisätyn henkilön tietoihin järjestelmässä. Koska kuntoutussuunnitelma on osa henkilön pakollisia tietoja suunnitelman tietoja voi jälkikäteen päivittää, mutta ei poistaa.



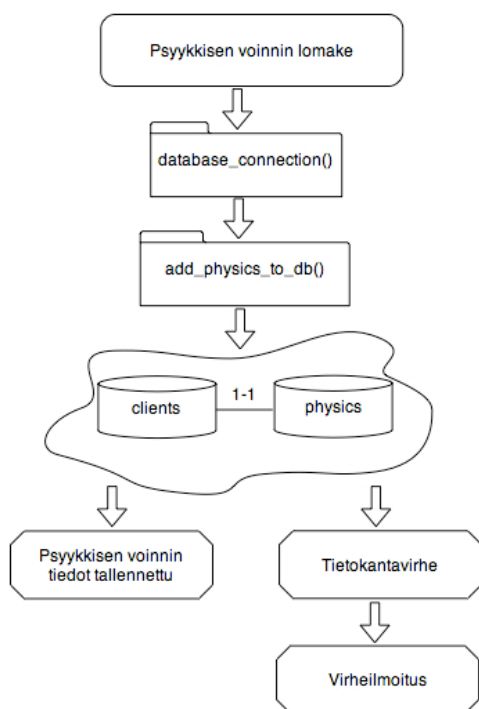
Kuvio 9. Kuntoutussuunnitelman lisääminen järjestelmään prosessikaaviona

Kentän nimi	Tyyppi	Pituus
id	int	11
client_id	int	11
resources	text	
targetlong	text	
targetshort	text	
tricks	text	
targetfuture	text	
date	varchar	50
present	int	11

Taulukko 6. Rehabilitation_plan-tietokantataulun rakenne

5.4.2 Psyykkinen vointi

Asiakkaan psyykkisestä voinnista tehdään kartoitus hänen tullessaan Pihlajakotiin. Psyykkisen voinnin lomake sisältää vapaatekstikentän, jota voi päivittää asiakastietojen alle.



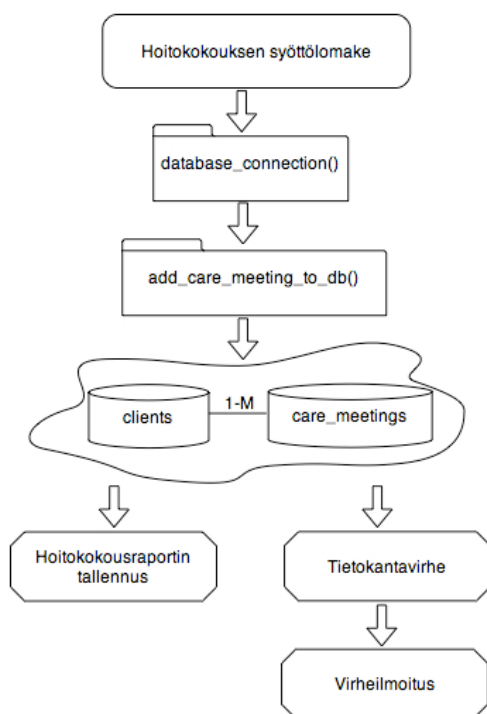
Kuvio 10. Asiakkaan psyykkisen voinnin lisääminen järjestelmään prosessikaaviona

Kentän nimi	Tyyppi	Pituus
id	int	11
client_id	int	3
free_text	longtext	
date	date	
author	varchar	30

Taulukko 7. Physics-tietokantataulun rakenne

5.4.3 Yksilötapaamiset

Asiakkaan tietojen alle on mahdollista tallentaa yksilötapaamisista koostetut raportit listattuina päivämäärän mukaan. Oletuksena käyttäjän alla avataan aina uusin tapaamisia koskeva raportti.



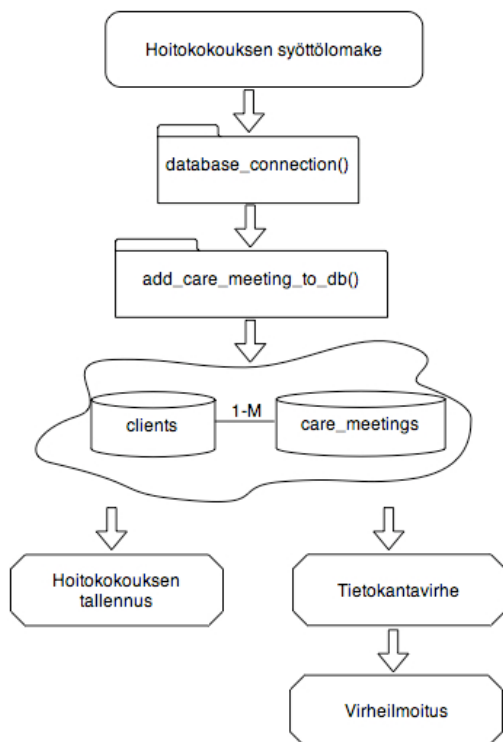
Kuvio 11. Asiakkaan yksilötapaamista koskevan raportin tallentaminen järjestelmään.

Kentän nimi	Tyyppi	Pituus
id	int	11
client_id	int	3
free_text	longtext	
date	date	
author	varchar	30

Taulukko 8. Personal_meetings-tietokantataulun rakenne

5.4.4 Hoitokokoukset

Yksilötapaamisten tavoin asiakkaan tietojen alle on mahdollisuus tallentaa hoitokokouksista koostetut raportit listattuina päivämäärän mukaan. Oletuksena käyttäjän alla avataan aina uusinta kokousta koskeva raportti.



Kuvio 12. Asiakkaan hoitokokousta koskevan raportin tallentaminen järjestelmään.

Kentän nimi	Tyyppi	Pituus
Id	int	11
client_id	int	3
free_text	longtext	
Date	date	
Author	varchar	30

Taulukko 9. Care_meetings-tietokantataulun rakenne

5.5 Ryhmät

Yksi järjestelmälle asetetuista vaatimuksista on Pihlajakodin toimipistekohtaisten ryhmien ylläpito ja niiden raportoinnin yhtenäistäminen, sekä seuranta eri toimipisteiden kesken. Ryhmät listataan järjestelmässä toimipisteiden mukaan ja niihin voidaan liittää vapaasti Pihlajakodin järjestelmässä olevia asiakkaita.

5.5.1 Ryhmän lisääminen järjestelmään

Järjestelmän peruskäyttäjä voi lisätä uuden ryhmän järjestelmään. Uuden ryhmän perustamisen yhteydessä käyttäjän tulee syöttää järjestelmään Ryhmän nimi, toimipaikka, sekä halutessaan lyhyt kuvaus ryhmästä, jonka perusteella asiakas voi halutessaan liittyä kyseiseen ryhmään tai sitä voidaan suositella hänelle. Vapaaehtoisena tietona käyttäjä voi lisätä ryhmälle myös aloituspäivämäärän.

Uuden ryhmän lisääminen

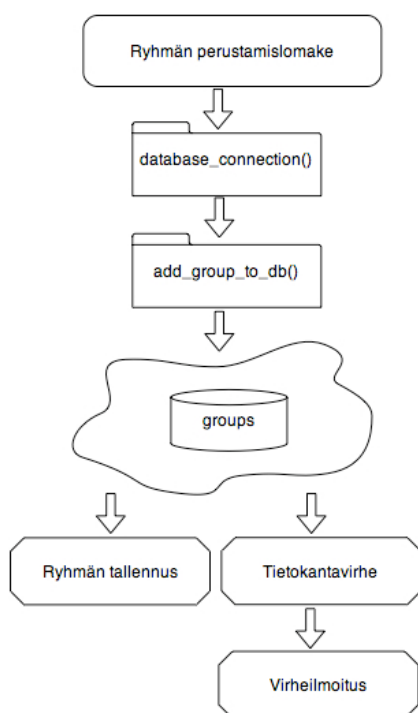
Ryhmän nimi

Ryhmän kuvaus

**Ryhmän aloituspvm.
(pp.kk.vvvv)**

Toimipaikka Valitse toimipaikka

Järjestelmäkuva 5. Uuden ryhmän lisääminen järjestelmään



Kuvio 13. Ryhmän lisääminen järjestelmään prosessikaaviona

Kentän nimi	Tyyppi	Pituus
id	int	11
groupname	varchar	255
groupdesc	longtext	
location_id	int	11
active	Int	1
startdate	date	
enddate	date	

Taulukko 10. Groups-tietokantataulun rake

Kentän nimi	Tyyppi	Pituus
group_id	int	11
id	int	11
comment	text	
author	varchar	255
date	char	255

Taulukko 11. Groups_comments-tietokantataulun rakenne

Kentän nimi	Tyyppi	Pituus
group_id	int	11
client_id	int	11

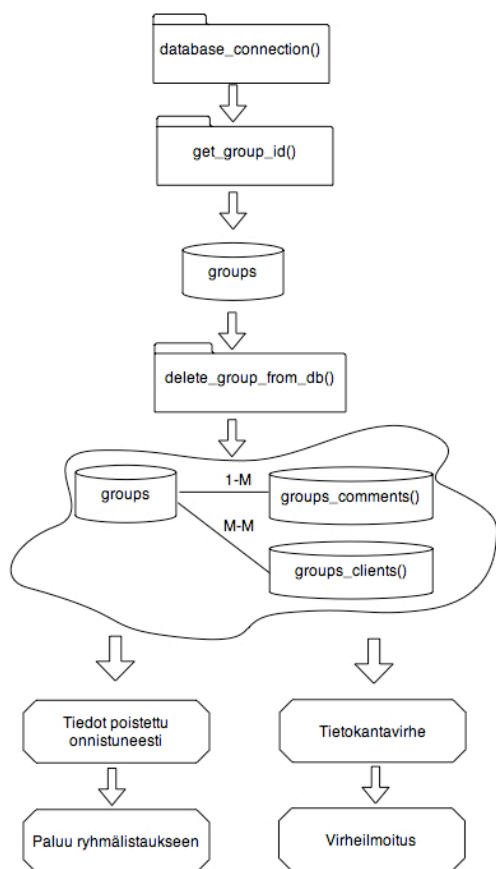
Taulukko 12. Groups_clients-tietokantataulun rakenne

5.5.2 Ryhmän poistaminen järjestelmästä

Järjestelmän peruskäyttäjä voi poistaa ryhmän järjestelmästä painamalla ”poista” -linkkiä ryhmien listauksessa. Sovellus varmistaa ryhmän poistamisen erillisellä ilmoituksella. Poistetun ryhmän mukana poistuvat myös kaikki ryhmäkohtaiset merkinnät, kommentit ja viittaukset ryhmässä olleisiin jäseniin.

RYHMÄT			
Testiryhmä3	Tämä on testiryhmä	Ryhmän tiedot	poista

Järjestelmäkuva 5. Ryhmälistaus



Kuvio 14. Ryhmän poistaminen järjestelmästä prosessikaaviona

5.5.3 Asiakkaan lisääminen ryhmään

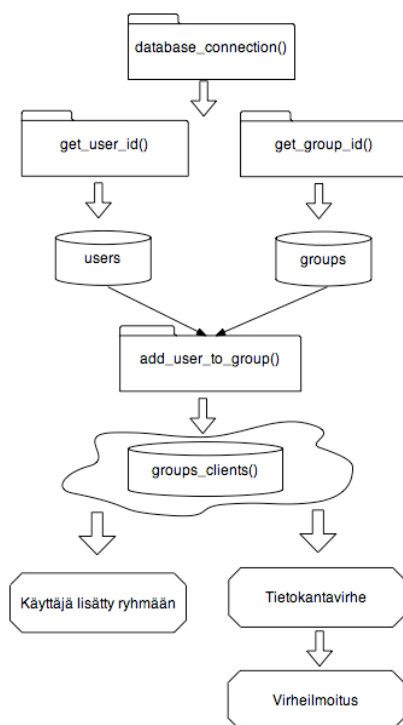
Järjestelmän peruskäyttäjä voi lisätä asiakkaan perustettuun ryhmään. Asiakkaan lisääminen tapahtuu valitsemalla halutun asiakkaan pudotusvalikosta.

Testiryhmä3

Tämä on testiryhmä

	Muokkaa selite-kentän sisältöä
Kokoontumispaikka	Puolarkuja
Ryhmän jäsenet	<div>Valitse lisättävä asiakas ▾</div> Lisää asiakas ryhmään
Ryhmäkohtaiset kommentit (viimeiset 3 kommenttia)	
10.02.10 - testiadmin kirjoitti:	
	Tämä on testikommentti
	poista kommentti
	Näytä kaikki kommentit
	Lisää kommentti

Järjestelmäkuva 6. Ryhmän tiedot



Kuvio 15. Asiakkaan lisääminen ryhmään prosessikaaviona

5.5.4 Ryhmäkohtaiset kommentit

Pikonetin peruskäyttäjä voi lisätä ryhmän tietoihin kommentteja, jotka koskevat koko ryhmää tai vain yhtä ryhmän henkilöä. Kommentit näkyvät sovelluksen käyttäjälle ja kertovat esimerkiksi jos ryhmän jäsen on jättänyt saapumatta ryhmään tai asiakas on käyttäytynyt ryhmässä sopimattomasti. Kommenttien määrää ei ole määriteltä, mutta ryhmän etusivulla näytetään vakiona kolme viimeisintä kommenttia navigoinnin helpottamiseksi

Ryhmäkohtaiset kommentit (viimeiset 3 kommenttia)

10.02.10 - testiadmin kirjoitti:

Tämä on testikommentti
poista kommentti
Näytä kaikki kommentit

Piilota kommentti-kenttä

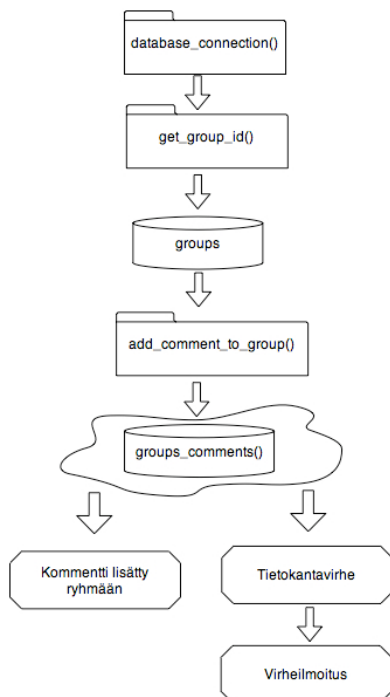
Kommentti koskee ▾
 Kommentti koskee

 Koko ryhmää

 Johan Strömberg

Tallenna kommentti tietokantaan

Järjestelmäkuva 6. Kommenttien lisääminen ryhmän tietoihin



Kuvio 16. Kommenttien lisääminen ryhmään prosessikaaviona

5.5.5 Ryhmän aktiivisuuden määrittäminen

Ryhmän tilaa on mahdollista muuttaa sen aktiivisuuden mukaan. Ryhmän pystyy poistamaan aktiivisten ryhmien listalta valitsemalla ”Ryhmä ei ole aktiivinen” -valinnan haluamansa ryhmän alta.

☒ Ryhmä ei aktiivinen

Järjestelmäkuva 7. ”Ryhmä ei aktiivinen”-painike

Valinnan jälkeen järjestelmä varmistaa käyttäjän valinnan ja ryhmä tallennetaan tietokantaan uudella aktiivisuus-arvolla (0). Tämän jälkeen ryhmän tiedot pystyy hakemaan ”Arkistoidut ryhmät”-listauksen alta. Ryhmän voi muuttaa takaisin aktiiviseksi poistamalla valinnan. Ominaisuus on käytännöllinen jos ryhmää ei haluta poistaa pysyvästi, mutta sitä ei tällä hetkellä toteuteta.

5.6 Raportit

Järjestelmän tehtävänä on tuottaa asiakasraportteja ja tallentaa ne arkistoon jatkokäsittelyä ja mahdollisia selvittelyjä varten. Järjestelmään on mahdollista tallentaa kahden tyyppisiä raportteja; Viikonloppu-, sekä vapaamuotoisia raportteja. Raportit tallennetaan järjestelmään niiden tyyppin mukaan ja niille on kaksi eri näkymää selaamisen helpottamiseksi.

5.6.1 Raportin lisääminen järjestelmään

Järjestelmän käyttäjällä on mahdollisuus luoda raportteja järjestelmään. Pakollisina tietoina toimivat raportin tyyppi, sekä päivämäärä. Itse raportti kirjoitetaan vapaatekstikenttään, eikä sen pituutta ole rajoitettu.

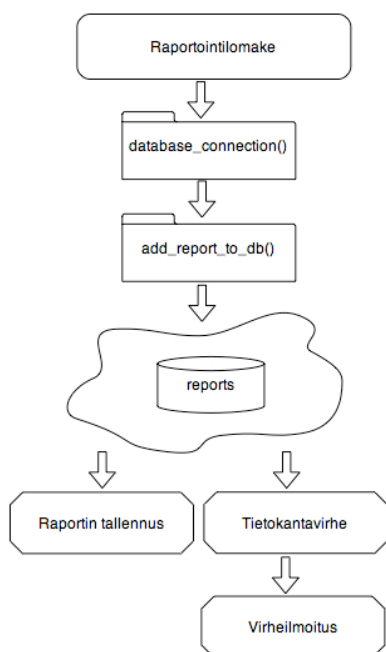
Uusi raportti

Raportin tyyppi Valitse raportin tyyppi ▾

Päivämäärä (pp.kk.vvvv)

Raportti

Järjestelmäkuva 8. Uuden raportin lisääminen järjestelmään



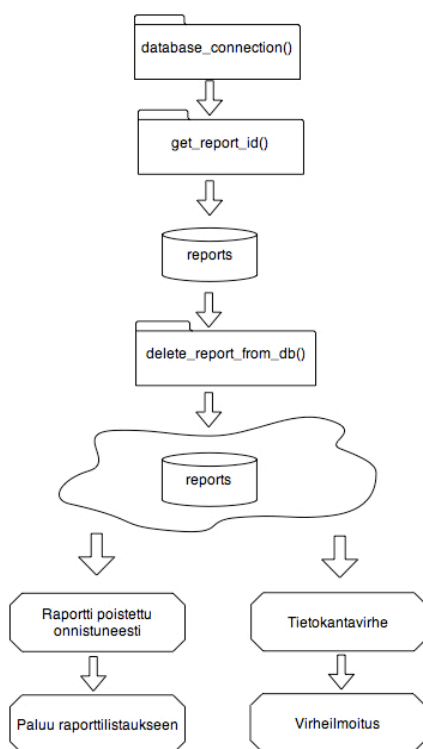
Kuvio 17. Raportin lisääminen järjestelmään prosessikaaviona

Kentän nimi	Tyyppi	Pituus
id	int	11
type	int	11
date	varchar	20
report	longtext	
location	int	11
subject	varchar	255

Taulukko 13. Reports-tietokantataulun rakenne

5.6.2 Raportin poistaminen järjestelmästä

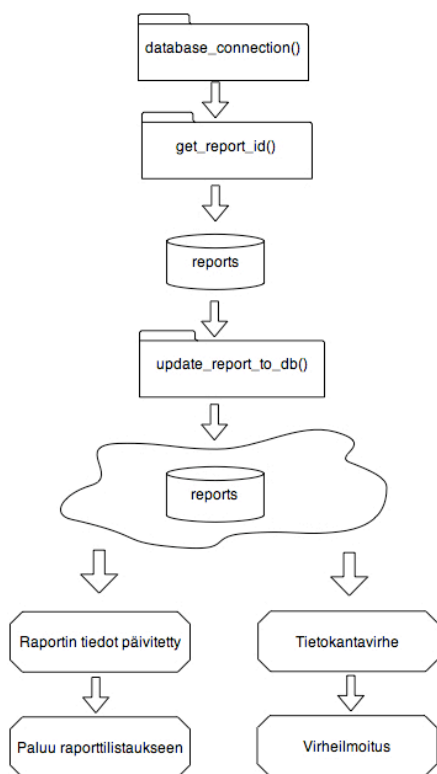
Järjestelmän peruskäyttäjä voi poistaa raportin järjestelmästä. Poistetun raportin myötä poistuvat myös muut siihen linkitetyt viitteet.



Kuvio 18. Raportin poistaminen järjestelmästä prosessikaaviona

5.6.3 Raportin muokkaaminen

Raporttien muokkaaminen tapahtuu raporttilistauksien alta. Raportin päivittäminen ei edellytä, että muokkaajana toimii raportin lisääjä.



Kuvio 19. Raportin tietojen päivittäminen prosessikaaviona

5.7 Muistutukset

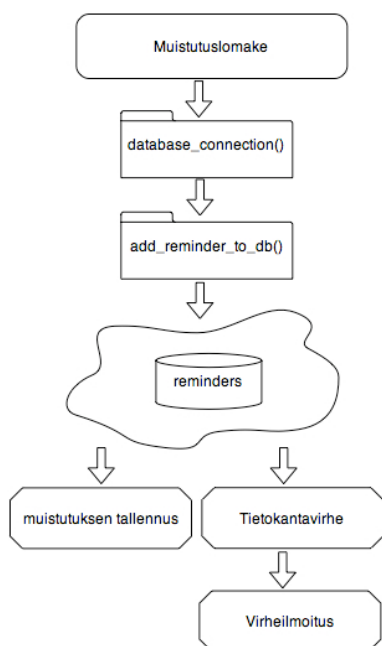
Järjestelmään on mahdollista lisätä käyttäjäkohtaisia muistutuksia. Muistutukset listataan niille kuuluvaan valikkoon siten, että lähin tuleva tapahtuma on ensimmäisenä listalla. Muistutukset poistuvat automaattisesti listalta, kun niille asetettu päivämäärä on umpeutunut ja arkistoituvat tietokantaan.

Käyttäjä pystyy lisäämään henkilökohtaisia muistutuksia järjestelmään täyttämällä muistutuslomakkeen. Pakollisia tietoja ovat aihe ja päivämäärä. Vapaatekstikentän pituutta ei ole rajoitettu.

Henkilökohtaiset muistutukset järjestelmässä

Aihe	<input type="text"/>
Päivämäärä	<input type="text"/>
Tehtävä	<div></div>
<input type="button" value="Tallenna muistutus"/>	

Järjestelmäkuva 9. Henkilökohtaisen muistutuksen lisääminen järjestelmään



Kuvio 20. Muistutuksen lisääminen prosessikaaviona

Kentän nimi	Tyyppi	Pituus
id	int	10
user_id	int	3
title	varchar	255
date	varchar	255
reminder	longtext	

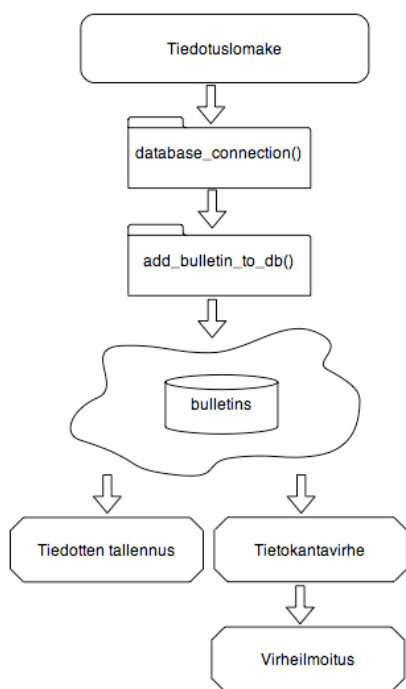
Taulukko 14. Reminders-tietokantataulun rakenne

5.8 Tiedotukset

Pääkäyttäjällä on mahdollisuus lisätä järjestelmään tiedotteita. Tiedotteet voidaan julkaista kaikille sovelluksen käyttäjille tai ainoastaan tietyssä toimipisteessä työskenteleville. Tiedotteet listataan ”Pihlajakodin yleiset tiedotteet” - valikon alle lisäämisjärjestyksessä.



Järjestelmäkuva 10. Pihlajakodin yleiset tiedotteet



Kuvio 21. Tiedotuksen lisääminen järjestelmään prosessikaaviona

Kentän nimi	Tyyppi	Pituus
id	int	10
title	varchar	255
bulletin	longtext	
location	int	5

Taulukko 15. Bulletins-tietokantataulun rakenne

6 Projektin yhteenveto

Pihlajakodin dokumentointijärjestelmän kehitystyö alkoi omalta osaltani jo lähes kaksi vuotta sitten. Oma osuuttani ennen valmiina oli kattava suunnitteludokumentti, joka oli tehty huolella, mutta kuten laajoissa projekteissa usein käy, dokumentissa määritellyt kokonaisuudet muuttuivat projektin aikana ja kasvattivat sen laajuutta.

Projektitapaamisissa Pihlajakodin henkilökunta toi esiin ominaisuuksia, jotka puuttuivat suunnitteludokumentista, mutta jotka olivat välttämättömiä projektissa olleiden toiminnallisuuksien kannalta. Tein projektin alussa sen virheen, että en rajannut toteutuksen laajuutta omalta kohdaltani tarpeeksi selkeästi, jonka johdosta toteutuksen piiriin tuli ominaisuuksia, jotka eivät olisi alkuperäisten suunnitelmien mukaan siihen enää kuuluneet.

Muuttuvan määrittelyn lisäksi haasteita projektille toi sovelluksen ulkoasun suunnittelu ja toteutus, joiden piti olla oma projektinsa. Tämä ei mielestäni kuitenkaan toiminut kuten sen olisi pitänyt. Projektin työmääriä laskiessani uskoin, että ulkoasu- ja käytettävyysprojekti todella olisivat omia kokonaisuuksiaan; Valmiin ulkoasun päälle tehdään varsinainen ohjelmakoodi. Tämän kaltainen työskentely on kuitenkin haasteellinen toteuttaa projekteissa, joissa ei ole selkeää aikataulua, eikä projektisuunnitelmaa ja osapuolet toimivat täysin erillään lähes koko projektin ajan. Ulkoasuvedokset kyllä toimitettiin ja niiden pohjalta käytiin keskustelua, mutta esimerkiksi html-pohjia en ikinä saanut.

Ulkoasun suunnittelulle tuotti haasteita myös Pihlajakodin pääkäyttäjän esteellisyys näön suhteen. Tämä oli tiedossa jo projektin alussa, mutta ensimmäisessä versiossa en osannut huomioida tätä asiaa tarpeeksi, jonka johdosta esimerkiksi sovelluksen värimaailma ei ollut mietitty riittävän pitkälle.

Edellä mainittujen lisäksi ongelmia tuotti myös Pihlajakodin palvelinympäristö, joka vaihtui kertaalleen projektin aikana. Tämä aiheutti sen, että projektissa oli katkoksia myös Pihlajakodin suunnalta, joiden aikana mitään ei voitu tehdä, koska testausta ei voitu suorittaa ja testidataa ei ollut saatavilla jatkokehitystä varten.

Toteutusprojektia ei olisi jälkeenpäin todettuna kannattanut mielestäni lähteä viemään eteenpäin oppilasvoimin, vaan teettää omina oppilastyönään suunnitelmat, kyselyt, sekä käyttäjätutkimukset ja graafisen ulkoasun määritykset, sekä toteutus. Varsinainen ohjelmointityö olisi ollut järkevää siirtää kokonaan ulkoiselle toimijalle tai vaihtoehtoisesti oppilaista koostuvalle projektiryhmälle, joka olisi toteuttanut projektin tarkoin suunnitelmin ja aikatauluin, työelämästä tuttujen metodien mukaan. Potilastietojen tallennusjärjestelmä on liian vastuullinen ja suuri kokonaisuus toteutettavaksi pelkästään oppilastyönä. Suunnittelu, toteutus, testaus, virheiden havainnointi, koodin tarkastaminen ja oikeellisuus ovat kaikki sellaisia työväiteitä, jotka eivät onnistu millään mittareilla, jos projektiryhmä koostuu vain yhdestä henkilöstä.

Ongelmista huolimatta projekti myös opetti paljon, huolimatta siitä, että olen työskennellyt ohjelmistoprojektien parissa koko opinnäytetyöprosessin ajan. Eniten projekti toi näkemystä asiakaslähtöiseen suunnittelutyöhön; Mitä kaikkea pitää huomioida, että projekti sujuu kitkatomasti ja asiakas saa juuri sitä, mitä on luvattu ja josta on sovittu. Sain myös käytännön ohjelmointityöhön uusia näkökulmia; Miten tehostaa toistuvan ohjelmakoodin käyttöä sekä vähentää toisteista tietoa tietokannassa että ohjelmoinnissa.

Jos projekti pitäisi aloittaa uudelleen puhtaalta pöydältä, tekisin monta asiaa toisin. Eniten muuttuisi suunnittelu- ja asiakaskommunikaatio. Paremman asiakaskommunikaation tuloksena työ olisi johdonmukaisempaa ja projektilla olisi selkeä alku- ja loppupiste. Koska kehitystyö on ollut käynnissä jo pitkään, se ei käytä uusimpia tekniikoita hyväkseen. Mikäli sovelluksen toteutustyö aloitettaisiin nyt, tekisin sen luultavasti jonkin valmiin CMS-alustan, kuten Drupalin päälle. Tämä säästäisi monta työvaihetta ja esimerkiksi arkkitehtuuri- ja tietokantasuunnittelu sekä tietokannat olisivat jo valmiina olemassa ja hyväksi havaittuja niiden toiminnan kannalta. Tätä kannattaa miettiä myös osana sovelluksen jatkokehitystä.

6.1 Tavoitteiden toteutuminen

Listasin työni alussa tavoitteita, joita tarkastelemalla voi arvioida projektin onnistumista omalta kohdaltani. Tavoitteet oli jaettu kuuteen, mielestäni tärkeään osa-alueeseen. Alla tarkastelen tavoitteiden täyttymistä.

6.1.1 Ylläpidettävyys

Tavoitteeni ohjelmakoodin ylläpidosta ei päässyt sille tasolle jolle olin sen ennen projektia suunnitellut vaan jouduin tekemään ohjelmoinnin suhteen tiettyjä kompromisseja. Ohjelmakoodi ei ole toisin sanoen mielestäni kaikilta osin parasta mahdollista. Se ajaa asiansa, mutta paremmalla suunnittelulla olisi saanut aikaan parempaa ja helpommin ylläpidettävää koodia.

Sisällön osalta ylläpidettävyystavoite mielestäni saavutettiin. Sisältöä on helppo lisätä, muokata sekä poistaa ja listoissa sekä muissa lomakkeiden komponenteissa olevat valintakenttien arvot tulevat dynaamisena datana tietokannasta kuten oli suunniteltu.

6.1.2 Luotettavuus

Sovelluksen luotettavuutta on vaikea arvioida tässä vaiheessa projektia, vähäisen testiaineiston ja tuotantodatan takia. Uskon kuitenkin sovelluksen olevan luotettava niillä mittareilla ja tietomäärillä joille sen toiminta on suunniteltu.

6.1.3 Tarkoituksenmukaisuus

Sovelluksesta tuli mielestäni tarkoituksenmukainen. Se ei tarjoa uusimpia tekniikoita tai erityisiä graafisia elämyksiä, mutta mielestäni sen ei tarvitsekaan. Sovellus tarjoaa alustan tiedon tallennukseen ja varastointiin ja toimii näiltä osin siis tarkoituksenmukaisesti.

6.1.4 Käytettävyys

Käytettävyys on nykyisessä sovellusversiossa mielestäni hyvä. Näin ei ollut ensimmäisessä versiossa jonka johdosta käyttöliittymä rakennettiin kokonaan uusiksi. Nykyinen versio tyydyttää minua, mutta mikä tärkeintä, se tyydyttää Pihlajakodin käyttäjiä.

6.1.5 Toiminnallisuus

Toiminnallisuuden osalta jouduin myös tekemään myönnytyksiä suhteessa suunnitelmiin. Kaikki suunnitellut toiminnallisuudet toteutettiin, mutta osa huomattavasti suunniteltua kevyempinä toiminnallisuuksien tai ulkoasun osalta. Toiminnallisuuksien toteutuksen voi laskea pie-noiseksi epäonnistumiseksi etenkin resurssivarauskalenterin suhteen, josta olin itse suunnitellut huomattavasti näyttävämpää graafisesti.

6.1.6 Jatkokehitys

Uskon, että sovelluksen jatkokehitys on funktioiden johdosta helppoa. Funktiot voi halutesaan viedä kohtuullisen helposti uuteen ulkoasuun tai kokonaan uuteen alustaan (esimerkiksi Drupal tai Liferay). Uudet ominaisuudet pystytään toteuttamaan omina kokonaisuuksinaan ilman, että nykyisiin funktioihin tarvitsee juurikaan koskea.

7 Sovelluksen jatkokehitys

Sovelluksen jatkokehityksen osalta kriittisimmäksi koen tietoturvan sekä tietokannan varmuuskopiointirutiinien kehittämisen. Nämä osa-alueet ovat välttämättömiä sovelluksen hyötykäytön kannalta tulevaisuudessa. tietokannan toimiessa samassa palvelimessa itse sovelluksen kanssa tietokanta on tärkeää saada kahdennetuksi toiselle palvelimelle, josta se on helppo palauttaa virhetilanteiden yhteydessä.

Muita tulevaisuuden projekteiksi jääviä osa-alueita ovat hakutoiminnot, henkilökuntaa ja työnseurantaa koskeva dokumentointi sekä toimintaterapiaa koskevat toteutukset.

Hakutoiminnot tullaan toteuttamaan joko vapaatekstihakuna, jolloin tietokanta pitää indeksoida tai vaihtoehtoisesti jokaiselle sovelluksen osa-alueelle toteutettavana ”täsmähakuna”, jolloin tietokanta joutuu huomattavasti pienemmälle rasitukselle, mutta vastaavasti ohjelmointityötä on tällöin huomattavasti enemmän. Sovellusta voi nykyisellään käyttää ilman hakua sen vähäisen tietosisällön johdosta. Tämä kuitenkin tulee muuttumaan tulevaisuudessa, joten toimiva, ja hyvin suunniteltu hakukone tulee olemaan tärkeässä osassa sovelluksen jatkokehitystä.

Henkilökuntaa ja työnseurantaa koskeva dokumentointi, sekä toimintaterapia-osiot tullaan toteuttamaan Pihlajakodin toiveiden mukaisesti mahdollisimman nopeasti. Etenkin toimintaterapialla on Pihlajakodissa tärkeä rooli ja tämän takia se on ensimmäisenä uusien toiminnallisuuksien listalla. Koska kyseessä on laaja kokonaisuus, jota ei ole suunniteltu eikä toteutettu lainkaan, tulisi toimintaterapia-osio mielestäni määritellä omaksi projektikseen. Edellisten lisäksi myös asiakkaan koostetietojen alle tullaan lisäämään uusia osioita. Tehtävää siis riittää ja on Laureasta kiinni, miten pitkälle sovelluksen kehitystyötä tullaan jatkamaan.

8 Lähteet

Kolehmainen Kauko. 2006. PHP & MySQL Teoriasta käytäntöön. Helsinki.

Hirsjärvi, Remes, Sajavaara. 2000. Tutki ja kirjoita. Helsinki.

9 Liitteet

Ohjelmakoodi (cd)

Käyttöohje

Liite 1 sisältää osan sovelluksen ohjelmakoodista ja esittää funktiokutsujen käytön sovelluksessa. Alla oleva koodikirjaston (add_functions) osa suorittaa uuden käyttäjän tallentamisen tietokantaan ja luo salasanan sekä sen salauksen. Sovelluksen lähdekoodi sekä MySQL-tietokanta löytyvät kokonaisuudessaan opinnäytetyöni liitteenä toimitetulta cd-levyltä.

```
<?php

function add_user()
{
    $Firstname=$_POST['FirstNametxt'];
    $Lastname=$_POST['LastNametxt'];
    $Workplace=$_POST['WorkPlaceTxt'];
    $Role=$_POST['RoleTxt'];
    $ID=$_POST['id'];
    $PermissionID=$_POST['idpermission'];
    $Email=$_POST['Emailtxt'];
    $Username1=substr($Firstname,0,2);
    $Username2=substr($Lastname,0,4);
    $Username=$Username2.$Username1;

    $Password = createRandomPassword();
    $PasswordMD5 = md5($Password);
    db_connect();

    $query="INSERT INTO users values($ID,$Firstname,$Lastname,$Workplace','','','0)";
    $query2="INSERT INTO permissions values($PermissionID,$ID,$Username,$PasswordMD5,$Role)";

    //Suoritetaan insert-komento

    mysql_query($query) or die ('Error, insert query failed');
    mysql_query($query2) or die ('Error, insert query failed');

    SendPasswordToUser($Email,$Username,$Password);
}

Tallennusfunktion kutsuminen sovelluksessa (add_to_db.php).

$Hae_ryhma=$_GET['action'];

if($Hae_ryhma=="adduser")
{
    add_user();
}
```

Käyttäjätietojen tallennuslomake

```
<FORM action="add_to_db.php?action=adduser" method="post">

    <?php db_connect() ?>
    <INPUT type="hidden" name="id" value="<?php get_last_no("users"); ?>">
    <INPUT type="hidden" name="idpermission" value="<?php
get_last_no("permissions"); ?>">

    <tr>
    <td width="100" span class='header_bold' bgcolor='#FFFFFF'>
    Uusi käyttäjä
    </td>
    </tr>

    <tr bgcolor='#FFFFFF'>
    <td width="200" span class='left_text'> Etunimi </td>
    <td>
    <INPUT type="text" name="FirstNameTxt">
    </td>
    </tr>

    <tr bgcolor='#FFFFFF'>
    <td width="200" span class='left_text'> Sukunimi </td>
    <td>
    <INPUT type="text" name="LastNameTxt">
    </td>
    </tr>

    <tr bgcolor='#FFFFFF'>
    <td width="200" span class='left_text'> Sähköposti </td>
    <td>
    <INPUT type="text" name="EmailTxt">
    </td>
    </tr>

    <tr bgcolor='#FFFFFF'>
    <td width="200" span class='left_text'> Toimipaikka </td>
    <td>
    <select name="WorkPlaceTxt">
    <option value="title">Valitse toimipaikka
    <?php
        db_connect();
        //Haetaan pudotusvalikon tiedot tietokannasta
        get_locations();
    ?>
    </select>
    </td>
    </tr>

    <tr bgcolor='#FFFFFF'>
    <td width="200" span class='left_text'> Rooli</td>
    <td>
    <select name="WorkPlaceTxt">
    <option value="title">Valitse rooli
```

```
<?php
    db_connect();
    //Haetaan pudotusvalikon tiedot tietokannasta
    get_roles();
?>
</select>
</td>
</tr>

<tr bgcolor='#FFFFFF'>
<td width="200" span class='left_text'></td>
<td>
<INPUT type="submit" value="Tallenna tiedot ja luo käyttäjätunnus">
</td>
</tr>

</FORM>
```


Liite 2 sisältää Pihlajakodin kirjaamissovelluksen käyttöohjeen. Ohje sisältää kaikki sovelluksessa olevat syöttö-, muokkaus- sekä poistamistoiminnot jaoteltuna toimintojen mukaan. Käyttöohje on toimitettu Pihlajakodin pääkäyttäjälle osana opinnäytetyötä.

PIKONET-KÄYTTÖOHJE

SISÄLLYS

1	Asiakastiedot	3
1.1	Asiakastietojen muokkaus sekä poistaminen järjestelmästä	3
2	Ryhmät	4
2.1	Uuden ryhmän lisääminen järjestelmään	4
2.2	Ryhmän tietosisällön ylläpito.....	5
2.2.1	Ryhmän selitekentän muokkaus.....	6
2.2.2	Jäsenten lisääminen ryhmään	6
2.2.3	Kommenttien lisääminen ja poistaminen ryhmän tiedoista.....	7
2.2.4	Ryhmän poistaminen	7
3	Raportit	8
3.1	Raportin lisääminen.....	8
3.2	Raporttien lukeminen ja muokkaaminen.....	8
3.3	Raportin poistaminen	9
4	Resurssivaraukset.....	9
4.1	Uusi varaus	9
4.2	Resurssivarausten selaaminen.....	10
5	ylläpitäjän hallintakonsoli	10
5.1	Käyttäjätietojen hallinnointi	11
5.2	Tiedotteet	12
5.3	Toimipaikkatietojen hallinta	12
5.4	Resurssien ylläpito	12

1 Asiakastiedot

Asiakastietoja ylläpidetään **Asiakkaat** välilehdellä.

Uuden asiakkaan lisääminen tapahtuu viemällä hiiren kursori **Asiakkaat** linkin päälle ja valitsemalla valikosta vaihtoehdon **Uusi asiakas**. Uuden asiakkaan lisäyslomake on jaettu kahteen osaan.

Asiakkaat	Ryhmät	Toimintaterapia	Raportit	Toimipisteet	Resurssit
Uusi asiakas	Uudet muistutukset	Strömberg Johan Testiasiakas Tero		ASIAKKAAT Asiakkaan tiedot Asiakkaan tiedot	
Pihlajakodin yleiset tiedotteet					
Testitiedote (03.04.2010)					

1.1 Asiakastietojen muokkaus sekä poistaminen järjestelmästä

Asiakastietojen muokkaus tapahtuu painamalla **Asiakkaat** linkkiä, jonka jälkeen pääikkunaan avautuu lista järjestelmään lisätyistä asiakkaista. Valitsemalla asiakkaan kohdalla **Asiakkaan tiedot** linkin käyttäjä pääsee esikatsелеmaan kyseisen asiakkaan tietoja.

Strömberg Johan	ASIAKKAAT Asiakkaan tiedot	poista
-----------------	-------------------------------	--------

Asiakastietojen muokkaus tapahtuu painamalla sivun ala-osasta **Muokkaa käyttäjän perustietoja** linkkiä. Muokatut tiedot tulevat voimaan painamalla **Tallenna muutokset** -painiketta lomakkeella.

Asiakastietojen poistaminen tapahtuu valitsemalla pääikkunasta poistettavan asiakkaan kohdalla **poista** - linkin. Järjestelmä varmistaa asiakkaan poiston, jonka jälkeen asiakastiedot poistetaan järjestelmästä.

Huomio! Poistettaessa asiakkaan tietoja tulee huomioida, että myös osa asiakkaaseen linkitetyistä muista tiedoista poistetaan pysyvästä järjestelmästä!

2 Ryhmät

Ryhmien tietoja ylläpidetään **Ryhmät** välilehden alla. Pihlajakodissa aktiivisesti toimivat ryhmät on tallennettu **Aktiiviset ryhmät** valikon alle ja vanhat tai tauolla olevat ryhmät **Arkistoidut ryhmät** valikon alle, josta ne on tarvittaessa palautettavissa aktiivisiksi.

	Ryhmät	Toimintaterapia	Raportit	Toimipisteet	Resurssit	Henkilökunta
set r	Aktiiviset ryhmät	AKTIIVISET RYHMÄT				
	Arkistoidut ryhmät	Tällä testiryhmällä on testaustarkoitus. Tämä on testiryhmä, jonka tarkoitus on demonstroida ryhmän toimintaa ja roolia järjestelmässä. Että näin ikkäänsä.				
	Uusi ryhmä					

2.1 Uuden ryhmän lisääminen järjestelmään

Ryhmän lisääminen tapahtuu viemällä hiiren kursori **Ryhmät** linkin päälle ja valitsemalla valikosta vaihtoehdon **Uusi ryhmä**.

	Ryhmät	Toimintaterapia	Raportit	Toimipisteet	Resurssit	Henkilökunta
set r	Aktiiviset ryhmät	AKTIIVISET RYHMÄT				
	Arkistoidut ryhmät	Tällä testiryhmällä on testaustarkoitus. Tämä on testiryhmä, jonka tarkoitus on demonstroida ryhmän toimintaa ja roolia järjestelmässä. Että näin ikkäänsä.				
	Uusi ryhmä					

2.2 Ryhmän tietosisällön ylläpito

Ryhmän tietoja ylläpidetään painamalla halutun ryhmän kohdalla **Ryhmän tiedot** linkkiä (ympyröity alla olevassa kuvassa).

	Ryhmät	Toimintaterapia	Raportit	Toimipisteet	Resurssit	Henkilökunta
set r	Aktiiviset ryhmät Arkistoidut ryhmät Uusi ryhmä	Testiryhmä Testiryhmä	AKTIIVISET RYHMÄT Tällä testiryhmällä on testaustarkoitus. Tämä on testiryhmä, jonka tarkoitus on demonstroida ryhmän toimintaa ja roolia järjestelmässä. Että näin ikkäänsä.			
						Ryhmän tiedot Ryhmän tiedot

Tämän jälkeen aukeaa ryhmän sivu, jossa on mahdollista muokata ryhmän tietoja, sekä aktiivisuutta.

Testiryhmä2

Tällä testiryhmällä on testaustarkoitus.

	Muokkaa selite-kentän sisältöä
Kokoontumisaika	Toimipaikka2
Ryhmän aloituspäivämäärä	
Ryhmän jäsenet	Tero Testiasiakas Johan Strömberg <input type="text"/> Valitse lisättävä asiakas
	Lisää asiakas ryhmään

Ryhmäkohtaiset kommentit (viimeiset 3 kommenttia)

Ei kommentteja

	Näytä kaikki kommentit
	Lisää kommentti

☐ Ryhmä ei aktiivinen

2.2.1 Ryhmän selitekentän muokkaus

Selitekentän muokkaus tapahtuu painamalla **Muokkaa selite-kentän sisältöä** linkkiä. Selite-kenttä päivittyy lomakkeelle painamalla **Päivitä kentän sisältö** painiketta. Selite-kentän muokkauksen voi perua painamalla **Piilota selitys-kentän sisältö** linkkiä.

Piilota selitys-kentän sisältö

Tämä on testiryhmä

Päivitä kentän sisältö

2.2.2 Jäsenten lisääminen ryhmään

Uusien jäsenten lisääminen ryhmään tapahtuu valitsemalla valikosta haluttu asiakas ja painamalla **Lisää asiakas ryhmään** linkkiä.

Ryhmän jäsenet	
	Tero Testiasiakas
	Johan Strömberg
	Valitse lisättävä asiakas
	Valitse lisättävä asiakas
	Tero Testiasiakas

2.2.3 Kommenttien lisääminen ja poistaminen ryhmän tiedoista

Ryhmäkohtaisten kommenttien lisääminen tapahtuu painamalla **Lisää kommentti** linkkiä, jonka jälkeen sivulle aukeaa kommenttikenttä. Kommentit voi merkitä koskemaan tiettyä asiakasta tai koko ryhmää. Kommentti tallennetaan **Tallenna kommentti tietokantaan** painikkeella, jonka jälkeen kommentit ovat luettavissa ryhmän sivulta. Ryhmän tiedoissa näytetään kolme viimeisintä kommenttia. Loput kommenteista saa näkyviin painamalla **Näytä kaikki kommentit** linkkiä.



Kommentit voidaan poistaa painamalla poistettavan kommentin alla olevaa **poista kommentti** linkkiä.

2.2.4 Ryhmän poistaminen

Ryhmän poistaminen tapahtuu painamalla pääikkunassa poistettavan ryhmän kohdalla **poista ryhmä** linkkiä. Järjestelmä varmistaa asiakkaan poiston, jonka jälkeen ryhmä poistetaan järjestelmästä.

Huomaa, että poistamalla ryhmän, poistuu myös kaikki siihen lisätyt tiedot.

3 Raportit

Raporttien ylläpito on jaettu kahteen osaan: Viikonloppu-, sekä muihin, aiheiltaan vapaisiin raportteihin. Raporttien ylläpito tapahtuu molemmissa tyypeissä valitsemalla **Raportit** valikosta haluamansa raportin tyyppin.

Asiakkaat	Ryhmät	Toimintaterapia	Raportit	Toimipisteet	Resurssit
Henkilökohtaiset muistutukset		03.04.2010	UUSI RAPORTTI Viikonloppuraportit Muut raportit	VIIKONLOPPURAPORTIT	

3.1 Raportin lisääminen

Raportin lisääminen tapahtuu valitsemalla **Raportit** valikosta **Uusi raportti** linkin, jonka jälkeen aukeaa sivu raportin tallentamista varten. Raportin tyyppi valitaan **Raportin tyyppi** valikosta. käyttäjän valitessa raportin tyyppiä **Muu raportti** on myös **Raportin aihe** pakollinen tieto lomakkeella. Valmis raportti lisätään painamalla **Tallenna tiedot** painiketta.

Uusi raportti

Raportin tyyppi

Päivämäärä (pp.kk.vvvv)

Raportti

3.2 Raporttien lukeminen ja muokkaaminen

Raportin lukeminen tapahtuu valitsemalla **Raportit** valikosta raportin tyyppi. Tämän jälkeen avautuvalta raporttilistalta painetaan halutun raportin kohdalla **Avaa raportti** linkkiä.

VIIKONLOPPURAPORTIT		
23.07.2009	Avaa raportti	poista raportti
31.01.2010	Avaa raportti	poista raportti
06.12.2009	Avaa raportti	poista raportti
24.01.2010	Avaa raportti	poista raportti

Raportin tietojen muokkaus tapahtuu painamalla raportin alaosassa olevaa **Muokkaa raporttia** linkkiä, jonka jälkeen raportti aukeaa ikkunaan.

Viikonloppuraportti (24.01.2010) / Testiadmin

Tämä on testiraportti.

[Muokkaa raporttia](#) | [Tulosta raportti](#)

3.3 Raportin poistaminen

Raportin poistaminen tapahtuu painamalla poistettavan raportin kohdalla **poista** linkkiä. Järjestelmä varmistaa raportin poiston, jonka jälkeen se poistetaan järjestelmästä pysyvästi.

4 Resurssivaraukset

4.1 Uusi varaus

Uusi resurssivaraus tehdään valitsemalla **Resurssit** valikosta **varaa resursseja** toiminto.

Toimintaterapia	Raportit	Toimipisteet	Resurssit	Henkilökunta
Varaaja <input type="text" value="KehvKa"/> Resurssi <input type="text" value="Valitse resurssi"/> Päivämäärä (pp.kk.vvvv) <input type="text"/> Aloitusaikajankohta <input type="text" value="08:00"/> Lopetusajankohta <input type="text" value="08:00"/> <input type="button" value="Varaa resurssi"/>			<input type="button" value="Varaa resursseja"/> <input type="button" value="Resurssivaraukset"/>	

Varaus tapahtuu valitsemalla halutun päivämäärän, sekä aloitus- ja lopetusajankohdat. Mikäli tietokannassa on varaus halutulle ajalle, järjestelmä ilmoittaa siitä varauksen yhteydessä.

4.2 Resurssivarausten selaaminen

Resurssivarausten selaaminen tapahtuu valitsemalla **Resurssi** valikosta **Resurssivaraukset** toiminnon, jonka jälkeen käyttäjällä on mahdollisuus syöttää päivämäärä, sekä valita haluamansa resurssi valikosta. Sovellus listaa kaikki valitun resurssin varaukset käyttäjän syöttämältä päivältä.

Toimintaterapia	Raportit	Toimipisteet	Resurssit	Henkilökunta
Päivämäärä (pp.kk.vvvv)		<input type="text"/>	Varaa resursseja	
Resurssi		Valitse resurssi ▼	Resurssivaraukset	
<input type="button" value="Näytä varaukset"/>				

5 ylläpitäjän hallintakonsoli

Sovelluksen ylläpitäjälle on muista käyttäjistä poiketen pääsy **Hallintakonsoliin**, josta ylläpidetään sovelluksen perustietoja. Hallintakonsolista voidaan ylläpitää sovelluksen käyttöoikeuksia, toimipaikkojen tietoja, sekä luoda uusia resursseja ja tiedotteita henkilökunnalle.

Hallintakonsoliin pääse valitsemalla **Asetukset** valikosta **Hallintakonsoli** toiminnon.

Raportit	Toimipisteet	Resurssit	Henkilökunta	Asetukset
				Yleiset asetukset
				Lisää muistutus
				Hallintakonsoli

5.1 Käyttäjätietojen hallinnointi

Uuden käyttäjän lisääminen tapahtuu valitsemalla hallintakonsolista, Hallinnointivalikosta **Käyttöoikeudet** linkin.



Tämän jälkeen aukeaa lista Pikonetin käyttäjistä.

Kehveli	Kalle	muokkaa	poista
Testaaja	Tero	muokkaa	poista
Strömberg	Johan	muokkaa	poista

[Lisää uusi käyttäjä](#)

Painamalla **Lisää uusi käyttäjä** linkkiä, aukeaa uuden käyttäjän perustietolomake.

Uusi käyttäjä

Etunimi	<input type="text"/>
Sukunimi	<input type="text"/>
Sähköposti	<input type="text"/>
Toimipaikka	Valitse toimipaikka <input type="button" value="v"/>
Rooli	Valitse Rooli <input type="button" value="v"/>
<input type="button" value="Tallenna tiedot ja luo käyttäjätunnus"/>	

Uusi käyttäjä, sekä käyttäjätunnus luodaan painamalla **Tallenna tiedot ja luo käyttäjätunnus** painiketta. Salasana tulostetaan ruudulle, jonka jälkeen se ei ole enää selkokielisessä muodossa, vaan pitää luoda uudelleen jos käyttäjä kadottaa sen.

Käyttäjä poistetaan järjestelmästä painamalla **poista** linkkiä halutun käyttäjän kohdalla.

5.2 Tiedotteet

Järjestelmän tiedotteita voi hallinnoida valitsemalla **Tiedotus** linkin, jonka jälkeen aukeaa listaus järjestelmässä olevista tiedotteista.

Testitiedote [muokkaa](#) [poista](#)

[Lisää uusi tiedote](#)

Uuden tiedotteen lisäys tapahtuu painamalla **Lisää uusi tiedote** linkkiä, jonka jälkeen avautuu tiedotteiden syöttösivu.

Uusi tiedote

Aihe
Tiedote

Tiedote koskee

Valitse toimipaikka 

Tallenna tiedote järjestelmään

Tiedotteen voi poistaa painamalla halutun tiedotteen kohdalla **poista** linkkiä.

5.3 Toimipaikkatietojen hallinta

Toimipaikkojen tietoja voi hallinnoida valitsemalla Hallinnointivalikosta **Toimipaikat** linkin, jonka jälkeen avautuu toimipaikkojen listaus.

Toimipaikka2	muokkaa	poista
Toimipaikka1	muokkaa	poista

[Lisää uusi toimipaikka](#)

Uuden toimipaikan voi lisätä painamalla **Lisää uusi toimipaikka** linkkiä, jonka jälkeen avautuu toimipaikan perustietolomake. Toimipaikan lisäämisen jälkeen se on käytettävissä järjestelmän lomakkeiden valikoissa.

Uusi toimipiste

Toimipaikka

Osoite

Puhelin

Luo uusi toimipiste

Toimipaikan voi poistaa valitsemalla **poista** linkin haluamansa toimipaikan kohdalla.

5.4 Resurssien ylläpito

Resursseja ylläpidetään valitsemalla Hallinnointivalikosta **Resurssit** linkin, jonka jälkeen aukeaa resurssilistaus.

Resurssi4	muokkaa	poista
Resurssi 3	muokkaa	poista
Resurssi 2	muokkaa	poista
Resurssi1	muokkaa	poista

[Lisää uusi resurssi](#)

Uusi resurssi lisätään painamalla **Lisää uusi resurssi** linkkiä, jonka jälkeen aukeaa resurssin syöttölomake. Resurssin tallentamisen jälkeen se on käytössä kaikissa sovelluksen osissa.

Uusi resurssi

Resurssin nimi

Toimipaikka

Luo uusi resurssi

Valitse toimipaikka ▼

Resurssi poistetaan painamalla halutun resurssin kohdalla **poista** linkkiä.